LiDAR Odometry by Deep Learning-based Feature Points with Two-step Pose Estimation

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

16-12-2021 / 20-12-2021

CITATION

Liu, Tianyi; Wang, Yan; niu, xiaoji; Le, Chang; Zhang, Tisheng; Liu, Jingnan (2021): LiDAR Odometry by Deep Learning-based Feature Points with Two-step Pose Estimation. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.17212967.v1>

DOI

[10.36227/techrxiv.17212967.v1](https://doi.org/10.36227/techrxiv.17212967.v1)

LiDAR Odometry by Deep Learning-based Feature Points with Two-step Pose Estimation

Tianyi Liu, Yan Wang, Xiaoji Niu, Le Chang, Tisheng Zhang and Jingnan Liu

Abstract—An accurate ego-motion estimation solution is vital for autonomous vehicles. In this paper, a method is proposed to preform LiDAR odometry utilizing bird's eye view of LiDAR data combined with deep learning-based feature point. Orthographic projection is applied to generate rasterized image by 3D point cloud. Then R2D2 neural network is employed to extract key points and compute their descriptors. Based on those key points and descriptors, a two-step matching and pose estimation method is utilized to keep them being tracked in a long distance with lower mismatch ratio compared with the conventional strategy. In the experiment, evaluation of the proposed algorithm on the KITTI training dataset demonstrates that the proposed LiDAR odometry can provide more accurate trajectories compared with handcrafted feature based SLAM algorithm. In addition, data collected by Velodyne VLP-16 is also evaluated by the proposed solution. Low-drift positioning RMSE of 4.70 m from about 5-km mileage shown in the result indicating that the proposed algorithm has generalization performance on low-resolution LiDAR.

Index Terms—LiDAR Odometry, bird's eye view (BEV), learning-based feature points, two-step pose estimation, feature tracking

I. INTRODUCTION

ATONOMOUS driving is the trend of intelligent transportation in this society, and high-level self-driving car has higher requirements for the positioning accuracy of vehicles. To obtain robust positioning results, multi-sensor fusion in navigation and localization has been widely researched in [1]–[4]. In the procedure of localization by multi-sensor fusion, making the most of the each single sensor can help improve the robustness and the accuracy of the entire system. GNSS as a main kind of global positioning method can provide high accuracy position information for its users under ideal environment. While it is always vulnerable [5], especially in the environment such as urban canyons, where the shadow of buildings and vegetation will cause the signal to be interrupted or seriously degraded. As an active ranging sensor, LiDAR can work under more severe conditions, and can provide accurate depth information directly. Additionally, working in area with pre-built map, LiDAR can provide reliable global positioning results. However, high-quality pre-built map is

hard to be accessed due to its high cost and the limitation of coverage. Hence, the research in LiDAR odometry is of great significance to the autonomous driving.

LiDAR odometry is a fundamental method for relative pose estimation. Data association and feature tracking in long distance are still obstacles for the improvment of accuracy. In conventional feature-based LiDAR odometry, feature points are always associated with the closest line or plane by the initial guess of pose [6], [7]. In data association with handcrafted descriptors, Random Sample Consensus (RANSAC) is implemented to determin the correspondences. While single threshold of distance in feature space inhibit these algorithms [8], [9] from improving the quality of correspondences and length of tracking. As a powerful tool for feature extraction, convolutional neural networks have made significant progress in recent years. These networks [10]–[13] are available to extract feature points on images with descriptors and have already outperformed the handcrafted features [14]–[16]. To alleviate these problems, a deep learning-based approach is employed for feature extraction and description. After that, these feature points are culled along with pose estimation by a two-step strategy to perform LiDAR odometry. The main contributions of this paper are as follows:

- 1) An accurate LiDAR odometry algorithm by deep learning-based feature point detection and description. The feature points are extracted from the BEV image of 3D LiDAR data. More accurate and robust key point association than handcrafted feature descriptor can be provided.

This work was supported by the National Key Research and Development Program of China under grant 2020YFB0505803 and the National Natural Science Foundation of China under Grant 41974024. (Corresponding authors: Xiaoji Niu, Tisheng Zhang.)

Tianyi Liu are with the GNSS Research Center of Wuhan University, Wuhan 430072, China (e-mail: liutianyi@whu.edu.cn)

Yan Wang, Xiaoji Niu, Le Chang, Tisheng Zhang, Jingnan Liu are with the GNSS Research Center, Wuhan University, Wuhan 430072, China (e-mail: wystephen@whu.edu.cn, xjniu@whu.edu.cn, changlesgg@whu.edu.cn, zts@whu.edu.cn, jnliu@whu.edu.cn)

Xiaoji Niu, Tisheng Zhang, Jingnan Liu are with the Artificial Intelligence Institute and also with the Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430072, China

- 2) A two-step feature matching and pose estimation strategy is proposed to improve the accuracy of key points association and length of feature tracking. The first step is to ensure the accuracy of data association, and the second step is to add more reliable feature points for long-range tracking.
- 3) The proposed method is evaluated by processing commonly used benchmark, KITTI dataset [17] and compared with the SLAM algorithm based on handcrafted feature points. The contribution of deep learning-based feature extraction and two-step pose estimation are verified by experiments respectively. In addition, generalization of the proposed algorithm has been proved by performing it on field test using low-resolution LiDAR, Velodyne VLP-16.

II. RELATED WORK

The ability of positioning and dead reckoning for autonomous vehicle is the fundamental guarantee for its autonomous driving capability. LiDAR odometry is an important means to provide positioning results when absolute positioning information is blocked. Currently, LiDAR odometry includes three main types, including geometry-based methods, deep learning-based methods and hybrid methods.

A. Geometry-based methods

The geometry-based methods takes geometric information in neighbor into account utilizing curvature to obtain feature points. LOAM [6], F-LOAM [7] and Lego-LOAM [18] extract feature points without descriptors by compute their curvature in neighbor. Uniform motion model is always adopted in the prediction of initial pose of current LiDAR frame. Then the feature point and the closest line or plane are associated by initial pose. Finally, the relative pose of LiDAR is estimated by minimizing the distance from point to line and point to plane. The information from adjacent scan lines is not fully utilized. In [19], [20], common landmarks extracted from infrastructures in urban areas such as poles and traffic boards are also employed for LiDAR scan matching. These types of method are constrained in some certain areas, and might be ineffective in country or highway. Since there are no descriptors for feature points or landmarks, mismatches can damage the performance of LiDAR odometry. In DLO [21] and DL-SLAM [22], height map of 3D LiDAR data is rasterized on XY-plane to generate image, and direct gray scale matching is utilized to compute translation and rotation. Ali [9] proposed a feature based LiDAR SLAM using rasterized image. ORB feature points are extracted on BEV of LiDAR data along with their descriptors. Then RANSAC algorithm is applied to associate these feature points. While, these approaches of feature extraction and feature point corresponding can be optimized in a further step for more accurate association and longer tracking.

B. Deep Learning based methods

This type of method generally preprocesses the point cloud by spherical projection to generate a multi-channel image.

Then the relative pose between LiDAR frames is estimated by end-to-end inference. LO-Net [23] takes the spherical projection image to perform end-to-end LiDAR odometry directly. DeepVCP [24] was trained on dense point cloud from Velodyne HDL-64 and accumulated them in 3 seconds and extracted 3D feature points for relative pose estimation. At the same time, some researchers are exploring unsupervised methods for learning-based LiDAR odometry, such as [25], [26], which reduces the data required for training effectively, but the accuracy is difficult to outperform the results by supervised learning. Point cloud preprocessed in form of spherical projection [23]–[28] makes it difficult to extract local feature accurately for low-cost scanning LiDAR as Velodyne VLP-16 due to its sparsity. To make the algorithm available to process the data of low-resolution LiDAR, the BEV image is employed in the proposed algorithm. In the meanwhile, deep learning-based methods have not been investigated on processing the BEV image of LiDAR data in LiDAR odometry, and it is worth being explored.

C. hybrid methods

In hybrid methods of LiDAR odometry, one type of them is to use traditional methods to extract feature points, and then regress relative pose of LiDAR frames by neural network. LodoNet [29] employs SIFT [14] for feature extraction on LiDAR spherical projection images, and then rearrange feature point correspondences into an $n \times 6$ vector (each LiDAR point contains 3-dimensional coordinates). Finally convolutional neural networks are trained for feature point culling, and regression of relative poses. The other stream of hybrid methods represented by DMLO [27], 3D3L [30] extract and match the feature points by means of deep learning on the spherical projection images of LiDAR frames, and then perform ICP [31] on the correspondences to obtain the solution by Singular Value Decomposition (SVD) or nonlinear optimization. While they keep the common flaw as the end-to-end method by taking the spherical projected image as input and it's not necessary to compute transformation by network modules.

In this paper, we combine feature extraction by deep learning neural network and nonlinear optimization to solve the rotation and translation in LiDAR odometry.

III. SYSTEM OVERVIEW

The scheme of the proposed algorithm shown in Fig. 1. The entire pipeline includes pre-processing module and module of LiDAR odometry. The red modules in the flowchart correspond to the main contributions of this paper. In the procedure of preprocessing, LiDAR point cloud is firstly segmented for retaining non-ground points to generate rasterized image. Then in the feature extraction stage, R2D2-Net is utilized for key point extraction and description. After that, feature points are stored in map database for further pose estimation and backend optimization.

Fig. 1: Workflow of the proposed LiDAR odometry algorithm. Showing the preprocess of LiDAR data and steps involved in mapping, pose estimation and optimization.

IV. PRE-PROCESSING

This part corresponds to the preprocessing module in Fig. 1. LiDAR mounted on a moving vehicle receiving reflections from ground all the time. These reflections do harm to pose estimation on XY-plane when point-to-point matching is applied. Consequently, LiDAR data is first segmented, keeping non-ground points in the environment to be further processed. Since only few points reflected from targets at a distance, in this paper, we crop the LiDAR BEV image along the x and y axis of LiDAR, leaving a square area of $[-37.5 \text{ m}, 37.5 \text{ m}]$ size. Then the retained LiDAR data is rasterized by orthographic projection, with resolution of 0.1 m, and the size of the processed image is 750×750 pixels. During the processing, the highest point z value in each grid is stored as the grayscale of the pixel, and its x, y value is also stored to provide information for subsequent pose optimization. This process is equivalent to linear transformation on the x and y coordinates of LiDAR frame. The projection transformation is indicated in (1).

$$\begin{bmatrix} u \\ v \end{bmatrix} = K \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (1)$$

For areas without LiDAR reflections, the gray level is uniformly set to 255. After the whole image is rasterized, the grayscale is linear stretched, and the height value of the original floating-point number is linearly transformed into the range of $[0, 255]$. In (2), the function *clamp* limits the output of it to a certain range $[h_{min}, h_{max}]$. And the *output* is the grayscale of pixel on the image in range $[0, 255]$, while *input* is the maximum height value in each grid, *a* and *b* are the coefficient and the bias of linear transformation.

$$output = a \cdot [clamp(input, h_{min}, h_{max}) + b]. \quad (2)$$

After converting the LiDAR BEV into an image, it is further processed by Gaussian blur to fulfill the holes in neighbor on the image.

V. LIDAR ODOMETRY

A. Feature Extraction and Matching

This part and the following ones correspond to the module of LiDAR odometry in Fig. 1. Data association in LiDAR odometry is curcial for accuracy, and the repeatability and reliability of feature points is of vital importance in data association. These attributes of feature points were considered the in desing of the R2D2-Net. It would neglect some areas with uniform textures in the optical image, such as the sky, ground, and water surface. Therefore, areas without targets would not be the region of interest to the neural network and no feature points would be extracted in these areas. The pre-processed LiDAR data is input into the R2D2 network as an image, and feature points are extracted, along with descriptors and response values. Similar to the response of the handcrafted feature point, the R2D2 feature point uses the product of repeatability and reliability from the network as the response. The higher the score is, the more reliable it is. After key point detection, those with the top-*k* strongest response is retained. Each feature point is described by a 128-dimensional normalized vector. Those descriptors satisfy that $\sum_{i=1}^{128} d_i^2 = 1$, where d_i is the *i*-th element in the description vector. Different from ORB descriptor, correlation is used to indicate the similarity between two key points, rather than hamming distance. An example of key points detected by R2D2 net is shown as Fig. 2. The background image is obtained by (1) and filtered by Gaussian kernel. It can be seen from the distribution of feature points that most of the points are located near points with large gradients in grayscale domain such as "corners" and "edges". In blank areas, only few feature points will be extracted.

Fig. 2: Key points detected by R2D2 net on KITTI dataset. The green circles are key points, and the background gray scale image is the BEV of LiDAR data.

B. Two-step Pose Estimation

In traditional LiDAR odometry or visual odometry, a uniform motion model is often utilized to make assumptions [6], [32]. According to the pose of the previous two frames, initial relative pose of current frame is obtained by linear interpolation as (3).

$$\tilde{T}_{k+1}^w = \hat{T}_k^w \left(\tilde{T}_k^{k+1} \right)^{-1} = \hat{T}_k^w \left(\hat{T}_{k-1}^k \right)^{-1}. \quad (3)$$

Where T_a^b is the relative transformation from frame a to frame b , including rotation and translation. Superscript w means world frame or map frame and subscript k is the timestamp. $\tilde{T}_k^{k+1}$ mean the initial guess of relative pose of frame k and frame $k+1$, and $\hat{T}_k^w$ is the posterior estimation of relative pose of frame k and map frame w . After extracting the feature points of each frame of image, in order to avoid violent searching and matching on the whole image, KDTree [33] is used in this algorithm to store the pixel coordinates of the feature points for searching their correspondences. When searching for the first time, since the assumption of a constant speed model is no always suitable for real vehicle motion, there will be a greater uncertainty in the pose extrapolation of the current frame. Therefore, a larger search radius is adopted. To avoid mismatches, a stricter descriptor distance threshold is set to confirm the correspondences. Then, each pixel coordinate of feature point is reprojected into local LiDAR frame by the projection matrix K to search for the nearest original point cloud. Their coordinates are involved into the cost function for pose optimization. By minimizing the distance of the reserved map point and the predicted map point by current pose of LiDAR as in (4), optimized pose can be computed.

$$g(R, t) = \sum_{i=1}^n \|p_i^w - (Rp_i^l + t_i)\|_2^2 \quad (4)$$

Where R and t is the rotation and translation of current LiDAR frame refer to the map frame w . p_i^w is the reserved map point, p_i^l is the i -th feature point in the current LiDAR frame corresponding to p_i^w .

C. Map Management

The position of the first frame is initialized as (0, 0) on XY-plane, and the heading is 0° . All feature points extracted by the network are recovered to the original LiDAR measurement as stated in Section IV, and their global coordinates are retained as map points. In the subsequent odometry calculation, for the feature points that are successfully matched, the observation information is associated with the corresponding map points, including the number of the LiDAR frame and the global pose of the LiDAR for backend optimization. When inserting a key frame, the LiDAR point cloud corresponding to all the feature points in the frame is transferred to the global coordinate system through the estimated LiDAR pose and marked as a map point for subsequent matching.

D. Key Frame Selection and Backend Optimization

When the number of tracked inliers is less than 100 points, or there are more than 5 frames between the last key frame and the current frame, then a key frame will be inserted. In the backend optimization, bundle adjustment is used to optimize the reserved 5 active key frames and the associated feature point observations. The position of the map point and the pose of the LiDAR key frames will be optimized in the same problem as several parameter blocks, by minimizing the distances between the feature points in LiDAR frames and the associated map points. In (5), $p_{i,j}^w$ is the map point of the j -th feature point observed in the i -th frame. R_i and t_i represent the rotation and translation of LiDAR in frame i refer to the global frame.

$$f(R_i, t_i, p_{i,j}^w) = \sum_{i=1}^m \sum_{j=1}^n \|p_{i,j}^w - (R_i p_i^l + t_i)\|_2^2. \quad (5)$$

In order to maintain the scale of the backend optimization, we reserve 5 active key frames in backend. Information of observations and frames out of the sliding window are removed. In general, when a key frame is going to be inserted, it will be judged whether the current frame meets the motion constraint: the closest key frame exceeds a certain distance threshold. If it is satisfied, the current key frame and the corresponding map point observation will not be inserted into map, otherwise, the earliest inserted active key frame and its map point observation will be removed. We apply ceres [34] library for non-linear optimization in this paper and choose Levenberg Marquardt as the optimizer.

VI. EXPERIMENTS

We have implemented the proposed LiDAR odometry algorithm and evaluated on the KITTI [17] dataset. The data is collected from three types of environments, i.e., country, urban and highway. The types of feature point cover a variety of scenes. The KITTI dataset provides 22 sequences of LiDAR data. 11 sequences of them from sequence 00 to sequence 10 are "training" data. The training data are provided with ground truth translation and rotation. In addition, data collected by low-resolution LiDAR, VLP-16 is also processed to validate the generalization of the proposed algorithm by the accuracy of relative motion estimation.

A. Evaluation of LiDAR Odometry

In this paper, 2D translation on XY-plane and heading angle are computed by the proposed algorithm. In the first experiment, we used the Absolute Trajectory Error (ATE) to evaluate the performance of the LiDAR odometry proposed. According to the method of EVO [35], the predicted trajectory is first aligned with the ground truth trajectory directly, and the z value of the estimated trajectories is regarded as 0 constantly. Then we perform the RMSE (Root Mean Square Error) and STD (Standard Deviation). The results are shown in TABLE I, the best RMSE and STD values are marked as bold font for each sequence.

The proposed algorithm outperforms the baseline method in some scenes with loop closure such as Seq. 06. As shown

TABLE I: Absolute trajectory error of our LiDAR odometry against LOAM and method in [36]. The unit of RMSE and STD is “m”.

Seq.	LOAM		Ali [36]		Our Method	
	RMSE	STD	RMSE	STD	RMSE	STD
00	13.89	6.37	7.66	3.15	6.08	3.48
01	47.91	29.44	21.08	9.58	5.99	1.79
02	19.88	5.86	16.62	9.36	12.47	4.60
03	3.68	2.08	1.65	0.74	1.20	0.45
04	2.73	1.47	0.94	0.47	0.28	0.13
05	4.39	1.99	4.48	2.52	1.13	0.49
06	3.68	1.92	3.51	1.31	0.66	0.24
07	1.82	0.69	3.5	1.66	0.50	0.22
08	15.02	6.83	11.67	2.16	2.65	1.31
09	7.94	3.06	6.31	2.79	6.13	3.46
10	7.18	3.61	5.28	2.95	2.72	1.25

in Fig. 3, scale errors of the trajectory appear in the result of ORB feature based laser SLAM [36] in Fig. 3b. Loop closure detection and optimization only made little contribution to reducing the error. To demonstrate the result of the baseline without their raw data, the right of the figure is adopted from the original literature [36], and the legend is modified, by placing “Ali’s method” in the figures.

Fig. 3: Trajectory of Seq. 06 in KITTI dataset by proposed LiDAR odometry and ORB feature point based laser SLAM. (a) Trajectory of Seq. 06 by our method. (b) Trajectory of Seq. 06 adopted from [36]. All trajectories shown above are aligned with ground truth.

This part of experiment proved the advantage of combining deep learning-based feature extraction and two-step pose estimation in an entire solution for LiDAR odometry.

B. Performance comparison between ORB and R2D2-Net

To demonstrate the contribution of deep learning-based feature extraction, we compared the multi-frame tracking performance of these two kinds of feature points. In the experiment, frame interval is set as 10 or 20 frames to demonstrate the robustness of feature extraction against viewing angle changes caused by long movement distance. We define the indication of performance as the average of matching inliers, the more points are tracked, the better performance it has. In this experiment, the number of key points extracted by R2D2 net is no more than the key points extracted by ORB and the raw LiDAR data is preprocessed in the same way. Similar

to [12], [13], we set 3 pixels as the radius to search for the correspondence.

Firstly, pixel coordinates of feature points in the k -th frame are transformed into the target frame ($k+10$ th or $k+20$ th frame) by ground truth and the projection matrix. Secondly, the nearest feature point in feature domain is regarded as the correspondences. If the distance in feature domain satisfied the threshold, it is considered as an inlier. Finally, the number of inliers is counted in each pair of frames to calculate the average inliers for each sequence.

The following figure shows the average of inliers for each sequence using two kinds of feature points at 10 and 20 frame interval respectively. The green triangles show the number of inliers detected by R2D2 net and the blue dots indicate the inliers detected by ORB. The result shows that R2D2 has a distinct advantage under the condition of 20-frame interval. Only in Seq. 06 and Seq. 10, the performance of R2D2 feature points is slightly worse when the frame interval is 10 frames. It means that more feature points extracted by R2D2 net could be tracked than those detected by ORB in the long run. It really helps to improve the accuracy by mapping and bundle adjustment in LiDAR odometry.

Fig. 4: Tracking inliers comparison of ORB and R2D2 feature extraction. (a) is the average number of inliers in each sequence with 10-frame interval. (b) is the average number of inliers in each sequence with 20-frame interval.

An example of feature matching by ORB and R2D2 are shown in Fig. 5. The dotted boxes show the mismatches of key points on the top, while matches by R2D2 feature points are consistent.

Fig. 5: Matches by ORB and R2D2 feature extraction. (a) shows matches by ORB feature points. (b) shows matches by R2D2 feature points.

One step further on analyzing the performance of ORB feature point and the feature points extracted by R2D2-Net.

There might be three reasons why the deep learning-based feature extraction method work well on the BEV image of LiDAR data.

- 1) Firstly, there are more rich and complex pattern in optical images than the BEV images of LiDAR. Trained by a large number of optical images, filters in the network can represent the feature space properly.
- 2) Secondly, 128-dimensional floating-point descriptors are inferred by the network, leading to more powerful description for those key points than the 256-bit descriptors of ORB feature.
- 3) Thirdly, the network constructed by multi-layer convolutional neural network has a large perception field to capture global feature to make feature points distinguishable.

C. Performance comparison between RANSAC and Two-step pose estimation

To validate the effectiveness of two-step strategy, contrast experiment between RANSAC and two-step pose estimation is performed. In the process of feature corresponding, 100 times of iterations are set in RANSAC pose estimation. Except feature corresponding process and motion estimation, other parts of the LiDAR odometry scheme are the same. The results are demonstrated in TABLE II. As mentioned in Section V-D, key frames are inserted when the number of tracked inliers is less than 100. The two-step method outperforms the RANSAC strategy in all sequences by less key frames and more observations for the feature points. As a result, an accurate odometry can be obtained with more independent observations for the same feature point according to [37], and the effectiveness can be proved.

TABLE II: Absolute trajectory error of two-step data association against RANSAC data association. The unit of RMSE and STD is “m”.

Seq.	R2D2+RANSAC		R2D2+Two step	
	RMS	STD	RMSE	STD
00	6.98	3.76	6.08	3.48
01	9.47	3.72	5.99	1.79
02	14.31	6.07	12.47	4.60
03	1.27	0.46	1.20	0.45
04	0.21	0.09	0.28	0.13
05	2.85	1.36	1.13	0.49
06	1.1	0.5	0.66	0.24
07	1.21	0.64	0.50	0.22
08	4.23	2.11	2.65	1.31
09	7.27	3.77	6.13	3.46
10	2.84	1.46	2.72	1.25

Tracking length of feature points is also analyzed by number of key frames in each sequence as the less key frames inserted, the longer feature points would be tracked. The contrastive results are shown in Fig. 6. It can be seen that, by two-step pose estimation, less key frames would be inserted in each sequence, indicating that more key points would be tracked longer. The blue histograms show the total number of LiDAR odometry computed by RANSAC pose estimation, and the orange histograms show the result of two-step pose estimation.

Fig. 6: Number of key frames inserted in each sequence by RANSAC and two-step pose estimation.

This part of experiment verified that two-step pose estimation strategy helps to improve the tracking distance of feature points as well as the accuracy of LiDAR odometry.

D. Evaluation on Velodyne VLP-16 dataset

In addition to the KITTI dataset, we also test the generalization of the proposed algorithm on low-resolution LiDAR data. We conducted data collection in Wuhan Research and Innovation Center, Wuhan City in January 2021. The street view of the park is shown in Fig. 7 and the scene is similar to the urban environment in KITTI dataset.

Fig. 7: Wuhan Research and Innovation Center from Baidu Street View.

The data collecting setup includes the quasi-navigation-level INS (Inertial Navigation System) LEADOR POS-A15 with gyroscope bias stability of 0.027°/h and accelerometer bias stability of 15 mGal. GNSS RTK was also provided in from of positioning results. This equipment is mounted on a Haval H6 SUV whose height is about 1.7 m. Extended Kalman Filter is utilized to obtain the reference trajectory in centimeter level by loose coupling. The setup of data collection system is shown in Fig. 8.

The data sequence length contains 8399 LiDAR frames, and last for 14 minutes. This park is nearly a square and the span is about 400 m in both the cross directions. Through EVO trajectory alignment and evaluation, the RMSE is 4.70 m and the STD is 1.55 m, on the same level as the result from KITTI dataset. There is no obvious drift even without loop closure optimization, which proves the effectiveness and generalization of the proposed algorithm. The result is shown in Fig. 9.

Fig. 8: Setup of data collecting system.

Fig. 9: Trajectory and position error plot for Wuhan Research and Innovation Center Dataset. The translation error is mapped onto the trajectory according to the colorbar.

Different from Velodyne HDL-64 whose field of view in vertical direction is 26.9° , Velodyne VLP-16 collects 16 scan lines of point cloud in the elevation angle range of $[-15^\circ, 15^\circ]$, 30° in total. Targets at different height will be perceived by these two types of LiDAR even if the sensors are mounted on the same height. As a robust feature extraction method with a proper pose estimating strategy, the performance won't degrade seriously. In this paper, we provide quantitative results to prove the effectiveness and generalization of the proposed method. Experiment conducted in Wuhan Research and Innovation Center shows that, the strategy for key points detection is practical on low-resolution LiDAR. The statistical values of LiDAR odometry tested on this dataset are on the same level of KITTI dataset.

VII. CONCLUSION

In this paper, we propose an algorithm of 2D LiDAR odometry based on BEV image. Deep learning-based feature extraction and two-step strategy are combined together for pose estimation. The RMSE of the trajectories shows that it outperforms the corresponding ORB based LiDAR SLAM on KITTI dataset even without loop closure in the proposed

method. The RMSE of positioning result reduces from 2% to 80% compared with the baseline method. The results of feature point tracking experiment show the effectiveness of deep learning-based feature extraction and two-step strategy. In addition, low-resolution LiDAR data from Velodyne VLP-16 was collected by us. The result is also presented to validate generalization of the proposed method. In the result of evaluation, the RMSE and STD of our dataset is 4.70 m and 1.55 m respectively in about 5 km-long mileage. Future research will focus on fusing more information from other sensors such as GNSS, IMU and wheel encoder to improve the accuracy and the adaptability in more complex environment.

APPENDIX I ODOMETRY RESULTS

In Fig. 10, we show all trajectories and their evaluations by EVO [35] on KITTI training dataset to provide an intuitive result.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (2020YFB0505803) and the National Natural Science Foundation of China (41974024). We want to make acknowledgment to Hailiang Tang for providing time-synchronizing device and his help in data collection. We also thank Suhan Huang for his work in the preparation of the experiments. At the same time, we want to appreciate Dr. Waqas Ali and Professor Peilin Liu from Shanghai Jiaotong University for their replies in the e-mail.

REFERENCES

- [1] L. Chang, X. Niu, T. Liu, J. Tang, and C. Qian, "Gnss/ins/lidar-slam integrated navigation system based on graph optimization," *Remote Sensing*, vol. 11, no. 9, p. 1009, 2019.
- [2] L. Chang, X. Niu, and T. Liu, "Gnss/imu/odo/lidar-slam integrated navigation system using imu/odo pre-integration," *Sensors*, vol. 20, no. 17, p. 4702, 2020.
- [3] K.-W. Chiang, G.-J. Tsai, Y.-H. Li, Y. Li, and N. El-Sheimy, "Navigation engine design for automated driving using ins/gnss/3d lidar-slam and integrity assessment," *Remote Sensing*, vol. 12, no. 10, p. 1564, 2020.
- [4] X. Li, H. Wang, S. Li, S. Feng, X. Wang, and J. Liao, "Gil: a tightly coupled gnss ppp/ins/lidar method for precise vehicle navigation," *Satellite Navigation*, vol. 2, no. 1, pp. 1–17, 2021.
- [5] Y. Du, J. Wang, C. Rizos, and A. El-Mowafy, "Vulnerabilities and integrity of precise point positioning for intelligent transport systems: overview and analysis," *Satellite Navigation*, vol. 2, no. 1, pp. 1–22, 2021.
- [6] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, Conference Proceedings.
- [7] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," *arXiv e-prints*, pp. arXiv-2107, 2021.
- [8] Ali, Waqas, P. Liu, R. Ying, and Z. Gong, "6-dof feature based lidar slam using orb features from rasterized images of 3d lidar point cloud," *arXiv preprint arXiv:2103.10678*, 2021.
- [9] Ali, Waqas and Liu, Peilin and Ying, Rendong and Gong, Zheng, "A life-long slam approach using adaptable local maps based on rasterized lidar images," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21 740–21 749, 2021.
- [10] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European conference on computer vision*. Springer, Conference Proceedings, pp. 467–483.
- [11] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Conference Proceedings, pp. 661–669.

Fig. 10: The estimated trajectories of each sequence in KITTI dataset. The positioning errors are mapped onto the trajectories.

- [12] J. Revaud, "R2d2: Reliable and repeatable detectors and descriptors for joint sparse keypoint detection and local feature extraction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshop*, Conference Proceedings.
- [13] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint detection and description of local features," *arXiv preprint arXiv:1905.03561*, 2019.
- [14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, Conference Proceedings, pp. 1150–1157.
- [15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, Conference Proceedings, pp. 404–417.
- [16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, Conference Proceedings, pp. 2564–2571.
- [17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving," in *Proc. CVPR*, Conference Proceedings, pp. 3354–3361.
- [18] T. Shan and B. Englot, "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)*. IEEE, Conference Proceedings, pp. 4758–4765.
- [19] A. Schaefer, D. Bscher, J. Vertens, L. Luft, and W. Burgard, "Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans," in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, Conference Proceedings, pp. 1–7.
- [20] T. Liu, L. Chang, X. Niu, and J. Liu, "Pole-like object extraction and pole-aided gnss/imu/lidar-slam system in urban area," *Sensors*, vol. 20, no. 24, p. 7145, 2020.
- [21] L. Sun, J. Zhao, X. He, and C. Ye, "Dlo: Direct lidar odometry for 2.5 d outdoor environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Conference Proceedings, pp. 1–5.
- [22] J. Li, J. Zhao, Y. Kang, X. He, C. Ye, and L. Sun, "DI-slam: Direct 2.5 d lidar slam for autonomous driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1205–1210.
- [23] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "Lo-net: Deep real-time lidar odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Conference Proceedings, pp. 8473–8482.
- [24] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvpv: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Conference Proceedings, pp. 12–21.
- [25] D. J. Yoon, H. Zhang, M. Gridseth, H. Thomas, and T. D. Barfoot, "Un-supervised learning of lidar features for use in a probabilistic trajectory estimator," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp.

2130–2138, 2021.

- [26] Y. Cho, G. Kim, and A. Kim, “Unsupervised geometry-aware deep lidar odometry,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Conference Proceedings, pp. 2145–2152.
- [27] Z. Li and N. Wang, “Dmlo: Deep matching lidar odometry,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Conference Proceedings, pp. 6010–6017.
- [28] R. Ambrus, V. Guizilini, J. Li, and S. P. A. Gaidon, “Two stream networks for self-supervised ego-motion estimation,” in *Conference on Robot Learning*. PMLR, Conference Proceedings, pp. 1052–1061.
- [29] C. Zheng, Y. Lyu, M. Li, and Z. Zhang, “Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation,” in *Proceedings of the 28th ACM International Conference on Multimedia*, Conference Proceedings, pp. 2391–2399.
- [30] D. Streiff, L. Bernreiter, F. Tschopp, M. Fehr, and R. Siegwart, “3d3l: Deep learned 3d keypoint detection and description for lidars,” *arXiv preprint arXiv:2103.13808*, 2021.
- [31] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, Conference Proceedings, pp. 586–606.
- [32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [33] R. B. R. Cousins and Steve, “3d is here: Point cloud library (pcl),” *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [34] S. A. Others and K. M. and, “Ceres solver.” [Online]. Available: <http://ceres-solver.org>
- [35] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” 2017. [Online]. Available: <https://github.com/MichaelGrupp/evo.git>
- [36] W. Ali, P. Liu, R. Ying, and Gong, Zheng, “A feature based laser slam using rasterized images of 3d point cloud,” *IEEE Sensors Journal*, 2021.
- [37] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

Tianyi Liu received the B.S. degree in electronic information engineering from Wuhan University, Wuhan, Hubei, in 2015 and the M.S. degree in signal and information processing from Chinese Academy of Science, Beijing, in 2018. He is currently pursuing the Ph.D. degree in communication and information system at Wuhan University, Wuhan, Hubei, China. His research interest includes deep learning, LiDAR SLAM and integrated navigation.

more than 30 patents. His research focuses on GNSS/INS integration, low-cost navigation sensor fusion, and the relevant new applications.

Xiaoji Niu received the bachelors (Hons.) and Ph.D. degrees from the Department of Precision Instruments, Tsinghua University, in 1997 and 2002, respectively. He did postdoctoral research at the University of Calgary, Canada, and worked as a Senior Scientist at SiRF Technology Inc. He is a Professor with the GNSS Research Center, Wuhan University, Wuhan, China. Dr. Niu is currently leading the Integrated and Intelligent Navigation Group (i2Nav). He has published over 100 academic articles and holds

Le Chang received the B.Eng. degree in Surveying Engineering from Wuhan University, Wuhan, China, in 2014. And he received an M.S. degree and Ph.D. degrees in geodesy and survey engineering in 2017 and 2021 from Wuhan University, Wuhan, China. His research interests focus on LiDAR SLAM and integrated navigation.

Tisheng Zhang received the B.Sc. and Ph.D. degrees in communication and information system from Wuhan University, Wuhan, China, in 2008 and 2013, respectively. He held a postdoctoral position at the Hong Kong Polytechnic University from 2017 to 2018. He is an Associate Professor with the GNSS Research Center, Wuhan University. His research interests focus on the fields of GNSS receiver and multi-sensor integrated navigation.

Yan Wang received the B.Eng. degree in Chemical Engineering and Technology from China University of Mining and Technology, Xuzhou, China, in 2016. And he received an M.S. degree in Computer Applied Technology from China University of Mining and Technology, Xuzhou, China, in 2019. He is currently pursuing a Ph.D. degree in GNSS Research Center, Wuhan University, Wuhan, China. His research interests focus on indoor Navigation, sensor fusion algorithm, and computer vision.

Jingnan Liu received the B.S. degree from Wuhan University, Hubei, China, in 1967 and the M.S. degree from the Wuhan University, in 1982, both in Geodesy. In October 1999, he became an Academician of Chinese Academy of Engineering. He is currently a Professor of GNSS Research Center, Wuhan University. His research interests are in Satellite Positioning and Navigation, etc.