



Deep Point Cloud Odometry: A Deep Learning Based Odometry with 3D Laser Point Clouds

Chi Li¹, Yisha Liu², Fei Yan¹, and Yan Zhuang¹(✉)

¹ The School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China

lichiduter@mail.dlut.edu.cn, {fyan,zhuang}@dlut.edu.cn

² Information Science and Technology College, Dalian Maritime University, Dalian 116026, Liaoning, China

liyisha@dlmu.edu.cn

Abstract. Deep learning-based methods have attracted more attention to the pose estimation research that plays a crucial role in location and navigation. How to directly predict the pose from the point cloud in a data-driven way remains an open question. In this paper, we present a deep learning-based laser odometry system that consists of a network pose estimation and a local map pose optimization. The network consumes the original 3D point clouds directly and predicts the relative pose from consecutive laser scans. A scan-to-map optimization is utilized to enhance the robustness and accuracy of the poses predicted by the network. We evaluated our system on the KITTI odometry dataset and verified the effectiveness of the proposed system.

Keywords: Pose estimation · 3D point clouds · Deep learning

1 Introduction

Laser odometry is widely used for autonomous driving and robot localization, which has been achieved great success. Classic laser odometry systems estimate poses by the laser registration methods, such as Iterative Closest Point (ICP) [1], Normal Distribution Transform (NDT) [11], and their variants [16, 17, 19]. Registration methods tend to be unreliable in some challenging scenarios, e.g., featureless places and motion with significant angular changes. Because of the sparsity of the point clouds caused by the low resolution of the laser scanner, the matching algorithm may not find the corresponding points or features, which may bring the drifts or even errors to the pose estimation.

Y. Zhuang—This work was supported in part by the National Natural Science Foundation of China under grant 61973049 and U1913201.

In recent years, the deep learning-based methods have attracted much attention in the research of geometry problems such as localization, relative pose estimation and odometry system. Many learning-based works are achieving state-of-the-art results in the field of visual odometry. Zhou et al. [24] presented a unsupervised training method to estimate the ego-motion from video. A novel Recurrent Convolutional Neural Network based VO system is proposed by Wang et al. [21] for dealing with the sequences data. [10] developed a unsupervised visual odometry which can estimate absolute scale and dense depth map simultaneously. Moreover, there are also a few laser odometry systems achieved in a data-driven fashion. [12] utilized the vanilla CNN (Convolutional Neural Network) for a laser odometry. Deep learning based 2D scan matching method is proposed by Li et al. [8] and [22] integrated deep semantic segmentation for the pose estimation.

Unlike regular data formats like images, the point cloud is unordered and sparse, which makes it difficult for the laser odometry to use the verified pipeline of the data-driven visual odometry. Some methods convert the point clouds into a structured representation for using the 2D or 3D convolution to extract the feature to estimate the ego-motion. [20] transformed the spare point clouds into the multi-channel dense matrix and employed the CNN to achieved the IMU assisted laser odometry. Qing Li et al. encoded the point clouds into the image-like formats by cylindrical projection and constructed a learning-based laser odometry. [9] DeepLO [2] proposed a deep LiDAR odometry via supervised and unsupervised frameworks using the regular point cloud representation. The projection lost the information of the original point cloud, so it is worth exploring to use point clouds to directly estimate odometry. Some works, like PointNet [14, 15], have made deep learning based on point cloud directly become a research hotspot.

In this paper, we propose a deep learning-based laser odometry using the point clouds as the input. Our main contributions are as follows: 1) We propose a scan-to-scan laser pose estimation network that directly consumes the irregular point clouds. 2) We use local map optimization to improve the robustness of network estimation, which makes up the laser odometry.

The rest of this paper is organized as follows. Section 2 shows an overview of the system. In Sect. 3, the proposed the system is presented. Experimental results are given in Sect. 4. The conclusions are drawn in Sect. 5.

2 System Overview

In this section, we briefly show our system, which is composed of a relative pose estimator and a local map pose optimizer, as shown in Fig. 1.

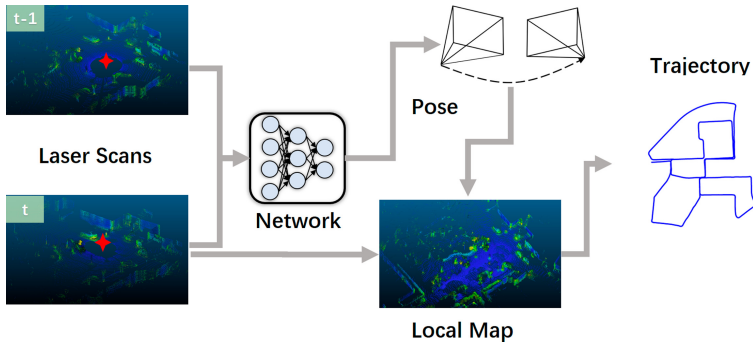


Fig. 1. System overview of the proposed deep point cloud odometry. The laser point cloud input into a network directly, and relative pose prediction of the network is further optimized by a local map matching.

The pose estimator is a PointNet-based CNN architecture, which is used to process the point cloud directly. It takes two consecutive point clouds as input and predicts the relative 6-DoF pose between them.

The pose optimizer is based on the ICP algorithm, which is used for point registration. The inputs of it are the relative pose predicted by pose estimator, the current point cloud, and the local map, and then it fine-tunes the pose by matching the point cloud to the local map.

Pose estimation only accumulating the scan-to-scan estimation tends to bring the errors over time, so the local map optimization is utilized to reduce the impact of cumulative errors.

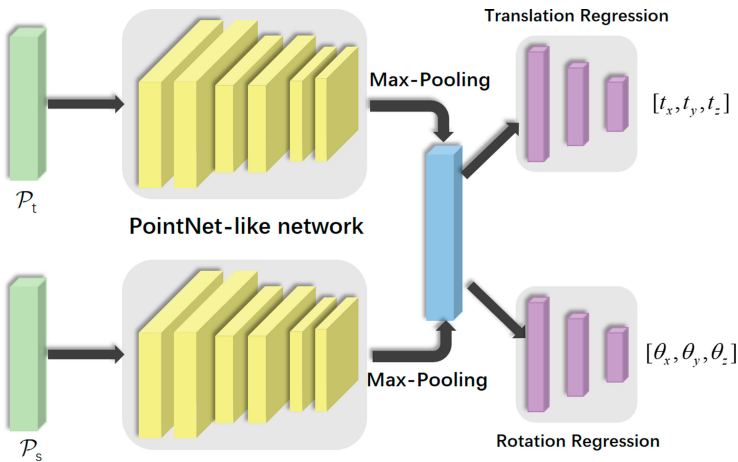


Fig. 2. Architecture of the network in proposed laser odometry system.

3 Pose Estimation with the Point Clouds

This section presents the proposed point clouds odometry composed of the deep pose estimation and local map pose optimization in detail.

3.1 Relative Pose Regression Through Convolutional Neural Networks

To estimate the relative pose of two consecutive laser scans, we train a network consisted of CNN-based feature extraction and a pose regression. The original points are used as the input of the network because they contain all the information which is needed to match.

The PointNet-like CNN architecture is employed to extract the feature of the point cloud, and then the features from different scans are combined and sent to the regressor to estimate the relative pose. As the Fig. 2 shows, the network takes two point clouds from consecutive laser scans: target point cloud \mathcal{P}_t and source point cloud \mathcal{P}_s as inputs and produce the 6-DoF relative pose: translation $\mathbf{t} = [t_x, t_y, t_z]^T$ and rotation in the form of Euler angle $\boldsymbol{\theta} = [\theta_{roll}, \theta_{pitch}, \theta_{yaw}]^T$ as output

$$\mathbf{t}, \boldsymbol{\theta} = \mathcal{F}(\mathcal{P}_t, \mathcal{P}_s). \quad (1)$$

We use \mathcal{L}_t and \mathcal{L}_r as the loss function to train the network.

$$\begin{aligned} \mathcal{L}_t &= \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2^2 \\ \mathcal{L}_r &= \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2^2 \end{aligned} \quad (2)$$

where $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\theta}}$ are the output of the network, \mathbf{t}^* and $\boldsymbol{\theta}^*$ are the ground truth. We use the ℓ_2 -norm in this work.

For training the network to learn the translation and rotation simultaneously, it is necessary to use a weight regularizer λ to balance the rotational loss with translational loss, because the scale and units between the translational and rotational pose components are different. To learn translation and rotation without including any hyperparameters, [6] presented a loss function that can learn the weight regularizer.

$$\mathcal{L}_{pose} = \mathcal{L}_t \exp(-s_t) + s_t + \mathcal{L}_r \exp(-s_r) + s_r \quad (3)$$

where s_t and s_r are the learnable parameters to regularize the scale between the translational and rotational losses.

3.2 Pose Optimization with Local Map

The pose optimization employs a scan-to-map matching with the geometry method to fine-tune the poses predicted by the network.

If the scan-to-scan matching creates errors, the rest of the trajectory will be affected by the errors. We propose maintaining a local map that can be used

to match the current scan for geometric constraints to modify the errors. The local map can improve the robustness of the odometry when some scan-to-scan matching creates errors.

An ICP is designed to register the current scan to the local map in the pose optimization, which takes the current scan, local map, and relative pose as input and computes the refined pose as output.

$$\Delta\hat{T} = \arg \min_{\Delta T} \frac{1}{2} \sum_{j=1}^N \|\Delta T p_j - p_{m(j)}\|_2^2 \quad (4)$$

where $p_j \in \mathcal{P}_s$ is the point in the source point cloud, $p_{m(j)} \in \mathcal{P}_m$ is p_i 's corresponding point in the local map, and $\Delta\hat{T}$ is the refined relative pose in the form of the special Euclidean group $SE(3)$ of transformations. The pose predicted by the network is used as the initial pose of the ICP. The ICP uses Eq. (4) as the cost function to match the scan to the local map iteratively and estimates the refined pose.

The local map contains historical point clouds over time, which needs to be maintained and updated. The local map updating comprises two steps: one step is removing the points that are outside the field of view from the local map which keeps the number of points in the local map not large, thereby map points culling can improve computational efficiency by reducing the computational complexity of searching for corresponding points, the other one is to add the points of the current scan to the local map, so that makes the local map has more extra feature points.

Table 1. Absolute translation errors (RMSE) of the test data from KITTI

| | 03 | 04 | 05 | 07 | 10 |
|-----------|----------|----------|-----------|----------|-----------|
| Ours | 4.873021 | 1.258067 | 5.221578 | 1.186617 | 14.592466 |
| LeGO-LOAM | 2.965074 | 0.511566 | 9.223725 | 0.921545 | 9.844019 |
| FGR+ICP | 3.173614 | 1.608129 | 29.039456 | 4.574337 | 14.550793 |

4 Experimental Results

In this section, we evaluate the performance of the proposed point cloud odometry. The network model is trained and tested by using publicly available datasets, KITTI odometry dataset [4]. The experimental results of local map optimization are also given in this section.

4.1 Implementation

We implemented the proposed system using PyTorch [13] and PyTorch Geometric [3], and trained the network with an NVIDIA RTX 2080ti. The optimizer employed the Adam Optimizer [7] to train the network with parameter $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate was initialized with 0.001 and decreased by 0.1 every 10 epochs until $1 * 10^{-6}$. The parameters s_t and s_r in Eq. (3) were set 0.0 and -2.0 respectively.

4.2 Dataset

The KITTI odometry dataset is a well-known public dataset of odometry benchmark. The dataset provides camera images, point clouds, Inertial Measurement Unit and other sensor data. We mainly use the point clouds which are captured by a Velodyne HDL-64E laser sensor. The dataset includes many driving scenarios, such as urban, streets, and highways. Sequence 00–10 of all 22 sequences of the dataset provide the ground-truth pose collected by the GPS/IMU sensor.

Our network was trained on sequences 00, 01, 02, 06, 08, and 09 and tested on sequences 03, 04, 05, 07, and 10. The point clouds inputted to the network were removed the grounds that may bias the evaluation results.

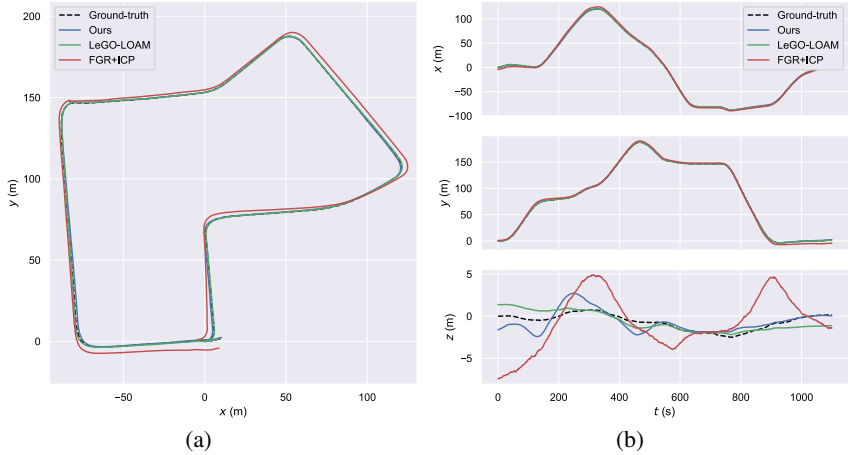


Fig. 3. Trajectories of KITTI sequence 07. The left figure (a) plots in XY plane. The right one (b) shows trajectories on the X, Y and Z axis respectively.

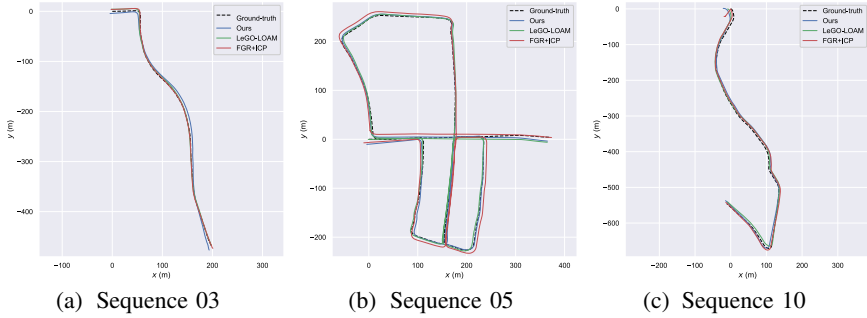


Fig. 4. Trajectories on test datasets.

4.3 Odometry Evaluation

We use averaged Root Mean Square Errors (RMSEs) of the pose errors to evaluate our system’s performance. The results of the evaluation of test datasets are shown in Table 1. The algorithms used to compare are LeGO-LOAM [18] and Fast global registration [23] with ICP fine-tuning, and all of the algorithms do not implement the loop closure detection. LeGo-LOAM is a state-of-the-art laser odometry system, which is the variant of LOAM, the top laser-based method in the KITTI odometry dataset. Fast global registration (FGR) is a global matching algorithm that is insensitive to an initial value and combines the local matching algorithm, ICP, to improve the pose estimation accuracy. We use the Evo tool [5], a python package for the evaluation of odometry and SLAM, to evaluate the experimental results of odometry.

Figures 3 and 4 show the predicted trajectories of the test datasets, in which the black dashed line is ground truth, the blue line is the proposed method, the green line is the LeGO-LOAM, and the red one is the FGR + ICP. It can be seen that the proposed system can provide nice results on the test datasets. This proves the proposed point cloud odometry is capable of learning to estimate the poses. From the details of the results in Table 1, we can see our system is not the best of all methods, so our algorithm also needs to improve performance, which can be achieved by training with more data.

4.4 Pose Optimization Evaluation

Figure 5 shows the comparisons of the pose predicted by the network with and without the local map optimization on the test dataset. The trajectories are on the top row, where the black dashed line is ground truth, the blue line is the result of after pose optimization, and the purple line is the output of the network. We utilized the box plot to show the error statistics on the bottom row. The top and bottom of the box are the 25th and 75th percentiles; the centerline is the median, and whiskers show the minimum and maximum errors.

From Fig. 5, it can be seen that the trajectories after optimization are more accurate than before optimization. Meanwhile, pose optimization also improves

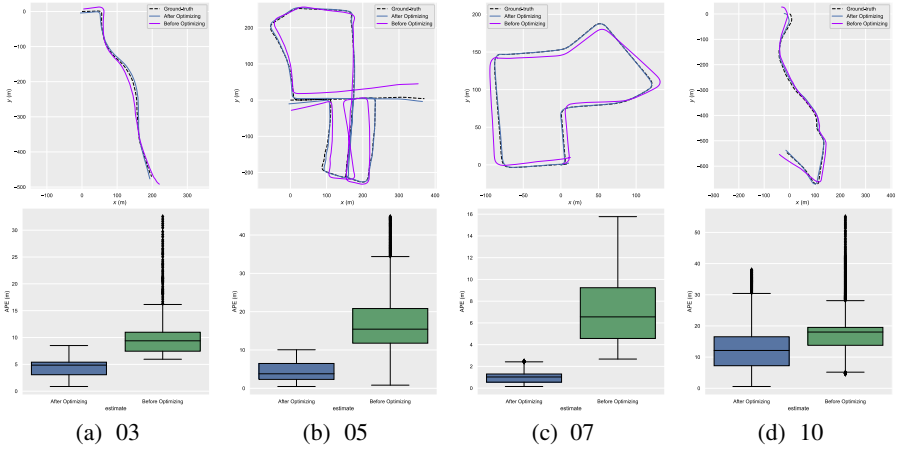


Fig. 5. Top row: trajectories of the pose estimation with and without local map optimization. Bottom row: box plot of the pose estimation without and with local map optimization which show the error statistics.

the system’s robustness. This proves that pose optimization is useful for the whole system and can help the network improve performance.

5 Conclusions

In this paper, we have presented deep point cloud odometry, a deep learning-based odometry with the point clouds. It estimated the poses by using the irregular point clouds directly and employed the local map optimization to improve the accuracy and robustness of odometry estimation. The results of the experiment showed that the proposed system could estimate the trajectories on the public dataset. In our future work, we plan to improve the generalization ability of the network to adapt to different resolution of the laser sensors and implement the deep learning-based method to map the point clouds.

References

1. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611, pp. 586–606. International Society for Optics and Photonics (1992)
2. Cho, Y., Kim, G., Kim, A.: DeepLO: geometry-aware deep lidar odometry. arXiv preprint [arXiv:1902.10562](https://arxiv.org/abs/1902.10562) (2019)
3. Fey, M., Lensen, J.E.: Fast graph representation learning with PyTorch geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361. IEEE (2012)

5. Grupp, M.: EVO: Python package for the evaluation of odometry and SLAM (2017). <https://github.com/MichaelGrupp/evo>
6. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6555–6564 (2017)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
8. Li, J., Zhan, H., Chen, B.M., Reid, I., Lee, G.H.: Deep learning for 2D scan matching and loop closure. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 763–768. IEEE (2017)
9. Li, Q., Chen, S., Wang, C., Li, X., Wen, C., Cheng, M., Li, J.: LO-Net: deep real-time lidar odometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8473–8482 (2019)
10. Li, R., Wang, S., Long, Z., Gu, D.: UnDeepVO: monocular visual odometry through unsupervised deep learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7286–7291. IEEE (2018)
11. Magnusson, M.: The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection. Ph.D. thesis, Örebro universitet (2009)
12. Nicolai, A., Skeeel, R., Eriksen, C., Hollinger, G.A.: Deep learning for laser based odometry estimation. In: RSS workshop Limits and Potentials of Deep Learning in Robotics, vol. 184 (2016)
13. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037 (2019)
14. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 652–660 (2017)
15. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5099–5108 (2017)
16. Segal, A., Haehnel, D., Thrun, S.: Generalized-ICP. In: Robotics: Science and Systems, Seattle, WA, vol. 2, p. 435 (2009)
17. Serafin, J., Grisetti, G.: NIPC: dense normal based point cloud registration. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 742–749. IEEE (2015)
18. 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. IEEE (2018)
19. Stoyanov, T., Magnusson, M., Andreasson, H., Lilienthal, A.J.: Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *Int. J. Robot. Res.* **31**(12), 1377–1393 (2012)
20. Velas, M., Spanel, M., Hradis, M., Herout, A.: CNN for IMU assisted odometry estimation using velodyne LiDAR. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 71–77. IEEE (2018)
21. Wang, S., Clark, R., Wen, H., Trigoni, N.: End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robot. Res.* **37**(4–5), 513–542 (2018)
22. Wong, J.M., et al.: SegICP: integrated deep semantic segmentation and pose estimation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5784–5789. IEEE (2017)

23. Zhou, Q.-Y., Park, J., Koltun, V.: Fast global registration. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 766–782. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_47
24. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1851–1858 (2017)