PCLane: Accurate Lane Localization with LiDAR and abstract data

Aditya Mushyam*
KOS-TEK, Institue of Transport Systems
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
Braunschweig, Germany

Oliver Boettcher
KOS-TEK, Institue of Transport Systems
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
Braunschweig, Germany

Franz Andert
KOS-TEK, Institue of Transport Systems
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
Braunschweig, Germany

***Abstract*—Accurate Lane localization is one of the fundamental tasks for autonomous driving and trajectory planning for precise control. Deep learning models are more suitable for accurate and precise lane detection and segmentation considering their scope for generalizations especially in complex urban scenarios like junctions and intersections. This paper presents a deep learning framework for LiDAR point cloud segmentation using abstract map data. To train such deep learning-based computer vision models for lane segmentation it is very important to have large datasets of annotated data. Point cloud data annotation is, especially, very expensive and difficult due to their sparseness and complexity when compared to camera images. This paper highlights a self-supervised annotation and training strategy of deep neural network-based model using abstract High Definition OpenDRIVE maps to detect and segment lanes from the lidar point cloud data. PointNet [1] based architecture was used as the backbone of the deep neural network model and the LiDAR point clouds were collected using Velodyne LiDAR sensor of the DLR Experimental Autonomous Driving Vehicle ViewCar2.**

Keywords—Lane detection, Lane Localization, Point cloud segmentation, OpenDRIVE map, Self-supervised learning

I. INTRODUCTION

Lane detection and localization is one for the fundamental aspect and capabilities required for automated driving and autonomous vehicles. The capability to detect and localize on lanes is primarily through vision sensors i.e. camera, lidar or both used together in a fused setup. LIDAR (Light Detection and Ranging) has quickly evolved into an important and extensively adopted perception sensor for autonomous driving especially for autonomous vehicles operating at higher levels of autonomy. It also provides precise geometric information about the environment and is thus a part of the sensor suites of almost all self-driving cars. Annotating LIDAR point cloud data is complex and an expensive process as a single point cloud is usually sparse and of very low resolution when compared to dense high-resolution camera image pixels, thus it is harder to accurately localize and recognize objects from the voxels for manual annotations. In contrast to annotation of 2D images with bounding objects of recognized objects in a frame, it is very complex and time consuming to draw 3D bounding boxes or even point-wise labels on LIDAR point clouds, requiring more than 1.5 hour for each frame [2]. Since data are usually collected in large sequences in driving tests and measurements campaigns, the collective dataset might consist of millions of point clouds, with highly overlapping consecutive frames. This leads to repeated and large-scale annotations which are economically unfeasible, since deep learning models would require large scale annotated datasets for the model to train effectively and generalize well on different kind of roads and lane scenarios in both Urban and Rural operational domains.

To overcome the hurdle of annotating large point cloud datasets which could be impossible to address with human intervention, certain strategies based on semi-supervised and unsupervised learning were also attempted based on different scenarios. A LaserMix model based on student teacher network framework using unlabeled point cloud data to extract strong special features from point clouds generated for, a mix of laser beams from different lidar scans is proposed which is agnostic to lidar representations [3]. But such a framework might not be suitable for road based lidar datasets, since the model uncertainty influences the quality and accuracy of annotations. Object detection from the point clouds generated from the LIDAR sensor can also could be achieved through classical algorithms like clustering, filtering and sensor fusion. Sensor Fusion based strategies are first applied on image-based data using detection algorithms to label an image and transfer the labels back to the point cloud. To augment the sensor

*Corresponding author aditya.mushyam@dlr.de

fusion, a one click strategy to identify the target object and generate bounding box is also used [4]. This strategy can be used effectively used to transfer the learning of the model from camera space to point cloud but could be limited to drawing boxes on the voxel but not for accurate segmentation of the voxels in the point cloud of a correspond object or lanes.

A clustering-based approach to identify the objects could also be applied on the point cloud elements based on their similarities or Euclidian and non-Euclidian distances. Various methods like k-means [5] and Euclidean cluster extraction [6] are applied for feature extraction. k-means leads to clustering of voxels in the point clouds into k-clusters constrained by average distances [5, 7] and it was found to be effective on discarding redundant dense points from an unevenly distributed point cloud [5]. To speed up the process a fast Euclidean clustering based point cloud segmentation method was proposed which is applied as pointwise scheme rather than a cluster wise scheme [8]. There are also other classical frameworks like edge based methods [9] which is based on finding the boundaries of each object and region growing based method [10]. Often these classical algorithms are constrained in performance and accuracy due to their rigidness in implementation especially in complex road network scenarios with multiple lanes, junctions and intersections due their lacking in generalization and unsuitability for online point cloud segmentation implementation on streamed point cloud similar to the processing the camera frames for detection and segmentation.

Deep learning models, with significant recent progress, are more suitable for accurate and precise lane detection and segmentation especially for online implementations to process point clouds, similar to the processing of the camera feed with object detection and segmentation networks like Yolov7 [11], DeeplabV3 [12] etc. Similar applications of deep learning frameworks for lane detection and segmentation have been developed and analyzed for their performance. Deep learning networks for object detection by key point estimation [13] are developed further into point instance segmentation model for lane detection on camera data [14]. Lane detection based on segmentation of lane markings on camera data and augmented with the estimation of lane curvature performed using least square fitting [15]. Since monocular image yields poor performance in autonomous driving and to predict 3D lane layout the explicit relationship between point clouds and camera images are exploited to annotate the data using deep learning frameworks [16]. Transfer learning of models trained on rich image based data to point cloud detectors that can learn general features of different objects and connecting them to textual prompting in a supervised framework has also been attempted [17]. But these models require initial human labelling of camera data and are dependent on the accuracy of these labels. Therefore, professional taggers are necessary to verify and correct the pseudo labels to ensure annotation accuracy and quality.

To ensure reliable annotation and completely eliminating human intervention we propose a self-supervised point cloud annotation framework based on abstract HD maps to training deep neutral networks for lidar point cloud data segmentation.

II. LIDAR POINT CLOUD DATASET

A. *Point Cloud Annotation*

First, to generate high quality large data sets with accurately labelled point clouds, avoiding expensive and time-consuming annotations, we propose self-supervised framework to annotate and segment LIDAR point cloud data specially in the context of lane detection and segmentation for autonomous driving using abstract OpenDRIVE map. The ASAM OpenDRIVE is a standardized format for road network descriptions using extensible markup language (XML). An OpenDRIVE file contains highly accurate data describing geometry of roads, lanes and objects, such as road marks on the road, as well as features along the roads, like signals etc. The OpenDRIVE format can be used to describe both real and synthetic road networks.

For the development of the annotation framework the open drive maps and point cloud data generated by DLR in TAVF Hamburg and Braunschweig Schwarzenberg driving campaign under the aegis of project OKULAr is used. The data set contains the accurate OpenDRIVE maps of the road network in both scenarios along with the point clouds collected parallelly while surveying the road network to generate an OpenDRIVE description.

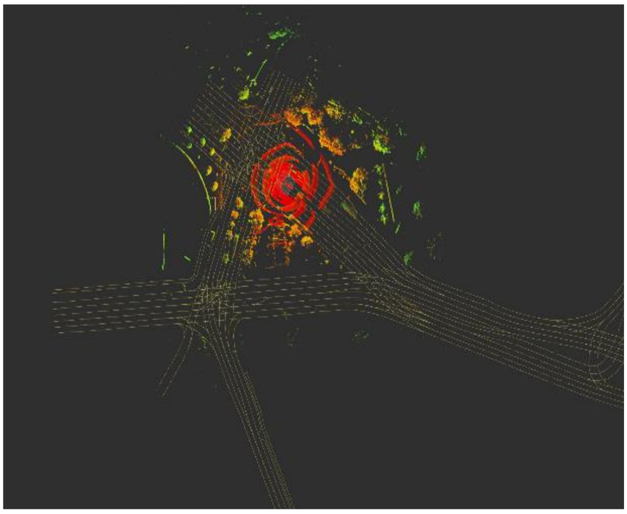

Fig 1: Localized Lidar Point Cloud data from TAVF campaign projected on an intersection represented in the OpenDRIVE map

B. *Annotation framework*

The self-supervised framework for point cloud data annotation is developed as a ROS package PClane using lidar point cloud data and OpenDrive maps in the context of project OKULAr. The self-supervised point cloud annotation training ROS package consisting of two data extractor nodes and a training node. The two data extractor nodes are for extracting point cloud data from the lidar scans collected during various measurement campaigns and road geometry data from OpenDrive maps. The lidar sensor data from the measurement campaigns are collated as a ROS bag file and the lidar data node accesses these individual lidar point cloud data frame and converts them to data file for training the deep learning model. Initially, the data from TAVF-Hamburg measurement campaign data was used for the model development and testing and this data is augmented with the Schwarzerberg measurement campaign data, for training and validation of the deep learning model architectures and to compare their performance. A section of the TAVF Hamburg dataset road network in OpenDRIVE format map representing an intersection with a localized instantaneous point cloud projected on to the OpenDRIVE map can be seen in Fig 1.

Algorithm for Self-supervised Annotation of Point Cloud data:

- The lane polygon used for the annotation of the point cloud data is extracted by the OpenDrive map node which loads the open drive map of the road section where the measurement campaign is performed from a region-based search in the rtree data structure used to store the map as seen in Fig. 2.
- The initial estimate of the vehicle position based on global positioning, a preliminary estimate of the local section of the road on the vehicle is currently present is determined.
- The OpenDrive map node, based on the initial position estimate, determines the information of the preceding and succeeding connecting road sections.
- The road sections thus determined also have the information of the lanes in the connecting road sections leading to the determination of the lane currently the vehicle in present.
- Since the geometry of the lanes of a road section are also integrated into open drive maps, the node extracts the information of the type and parameters of geometry of the lane from the preceding and succeeding road section.
- This geometric information of the lane polygon is transformed to the refence frame of the Ego vehicle and then used to annotate the point clouds.
- Based on these generated corresponding labels and respective annotated point clouds a deep learning-based segmentation architecture is trained for the detection of current lane detection and segmenting the voxels in the point cloud corresponding to the current lane

III. TRAINING AND EVALUATION

A. Lane Polygon at Intersections

Multiple lane polygons might be close to the position due scenarios like Intersections and Junctions as seen Fig 3. To extract the correct lane by selecting the correct lane polygon used to annotating the point cloud at the localized position in road network frame of reference, a Lane divergence score was used to rank the polygons and select the lane polygon containing he current ego vehicle position as defined in equation 1.

$$\text{Lane Divergence Score} = \sum_{k=1}^{n} dist(p^t, v^i) \qquad (1)$$

Where, p^t $is\ the\ ego\ position,$

v^i $are\ vertices\ of the\ polygon.$

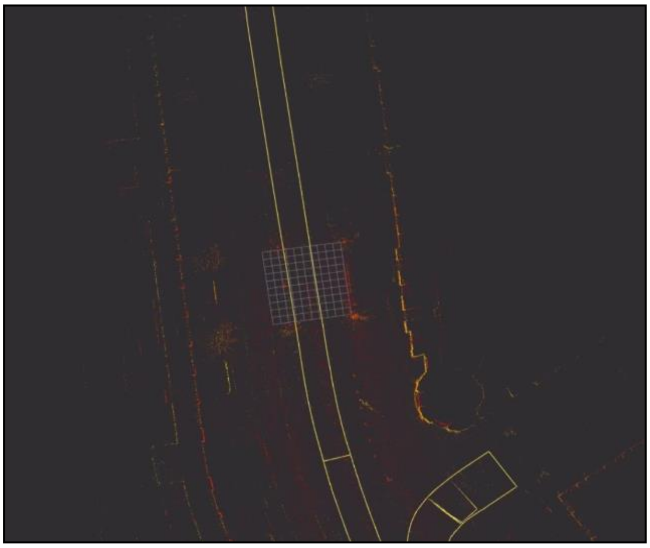

Fig 2: Lane Polygon extraction from OpenDRIVE map for the corresponding localized ego vehicle position in OpenDRIVE road network frame of reference.

Scenario of lane selection could also arise due to inaccuracies or erratic jumps in the global position and a more accurate global position could be achieved through GNSS and fused with inertial positioning system. The region still with multiple lanes extracted from OpenDRIVE map due to complex scenarios like junctions are handled with the ranking strategy using Lane Divergence Score in equation (1).

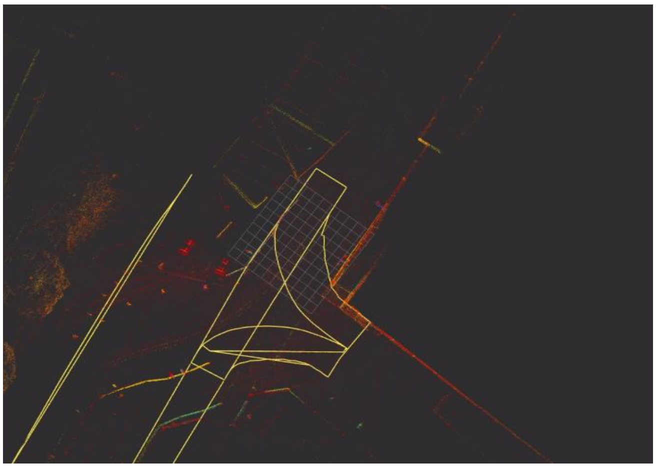

Fig 3: Lane Polygon selection at an intersection from OpenDRIVE map for the corresponding localized ego vehicle position in OpenDRIVE road network frame of reference.

B. PointNet Architecture as Backbone:

PointNet[1] architecture was used as the backbone for the deep earning model and the architecture consists of local and global feature extraction strategy. The classification network takes fixed number of points n as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for *k* classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. “mlp” stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net [1].

Initially the architecture was trained on road objects using the labeled data set of Semantic KITTI [18] consisting of different road objects like Car, Motorcycles etc and then by applying transfer learning these learnings were embedded in further training of the model on the lanes extracted from the open drive map and consisting in total of 12 classes.

TABLE I. RESULTS OF SEMANTIC SEGMENTATION

Point Net		
	Mean IOU	***Overall accuracy***
Baseline PointNet*	47.71	78.62
PCLane†	38.48	74.65

* Average IoU over 13 classes (structural and furniture elements plus clutter) and classification accuracy calculated on points [1].

† Average IoU over 12 classes consisting of Road Objects and lanes and classification accuracy calculated on points.

After the segmenting the point cloud and extracting the voxels corresponding to the current lane section corresponding to the current ego location in the ego frame of reference, the vehicle can be accurately localized as the distances are inherently captured in a 3D point cloud data. The lane geometry is characterized by the hull extracted from the segmented point cloud section, made of the outer most voxels and treating them as the current lane boundary for ego vehicle trajectory planning and control as required for autonomous navigation.

IV. Conclusion

In this work we propose a self-supervised frame work for point cloud annotation and lane segmentation based on abstract High definition OpenDrive Maps. A detailed strategy for Point cloud data annotation based on the pseudo labels extracted by projecting the point cloud into the road network frame of reference of the OpenDrive maps and node automation using the ROS based framework is presented. Pointnet Deep Neural Network architecture was employed and trained on the annotated point cloud data for segmentation of Road Objects and lanes. The segmentation performance of the network is comparted with PointNet architecture on ModelNET40 dataset. The performance was found to be on par with the respect to segmentation of point cloud data and eliminating completely the manual annotations of point clouds.

Acknowledgement: This work was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) within the research project “OKULAr” (Grant No. 19A22003C).

References

[1] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2016. PointNet*: Deep Learning on Point Sets for 3D Classification and Segmentation*.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. *The Cityscapes Dataset for Semantic Urban Scene Understanding*

[3] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. 2022. *LaserMix for Semi-Supervised LiDAR Semantic Segmentation.*

[4] Bernie Wang, Virginia Wu, Bichen Wu, and Kurt Keutzer. 2019. *LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking*.

[5] Bao-Quan Shi, Jin Liang, and Qing Liu. 2011. Adaptive simplification of point cloud using k-means clustering. *Computer-Aided Design* 43, 8, 910–922. DOI: https://doi.org/10.1016/j.cad.2011.04.001

[6] Radu B. Rusu. 2010. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. *Künstl Intell* 24, 4, 345–348. DOI: https://doi.org/10.1007/s13218-010-0059-6

[7] Rui Xu, Jie Xu, and Donald C. Wunsch. 2010. Clustering with differential evolution particle swarm optimization. In *IEEE Congress on Evolutionary Computation*. IEEE, 1–8.

[8] Yu Cao, Yancheng Wang, Yifei Xue, Huiqing Zhang, and Yizhen Lao. 2022. FEC: Fast Euclidean Clustering for Point Cloud Segmentation. *Drones* 6, 11, 325. DOI: https://doi.org/10.3390/drones6110325.

[9] A. D. Sappa and M. Devy. 2001. Fast range image segmentation by an edge detection strategy. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE Comput. Soc, 292–299. DOI: https://doi.org/10.1109/IM.2001.924460.

.[10] Ming Huang, Pengcheng Wei, and Xianglei Liu. 2019. An Efficient Encoding Voxel-Based Segmentation (EVBS) Algorithm Based on Fast Adjacent Voxel Search for Point Cloud Plane Segmentation. *Remote Sensing* 11, 23, 2727. DOI: https://doi.org/10.3390/rs11232727.

[11] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan M. Liao. 2023. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7464–7475.

[12] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. 2020. *Image Segmentation Using Deep Learning: A Survey*.

[13] Zechen Liu, Zizhang Wu, and Roland Tóth. 2020. *SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation*.

[14] Yeongmin Ko, Younkwan Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. 2020. *Key Points Estimation and Point Instance Segmentation Approach for Lane Detection*..

[15] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end Lane Detection through Differentiable Least-Squares Fitting.

[16] Fan Yan, Ming Nie, Xinyue Cai, Jianhua Han, Hang Xu, Zhen Yang, Chaoqiang Ye, Yanwei Fu, Michael Bi Mi, and Li Zhang. ONCE-3DLanes: Building Monocular 3D Lane Detection.

[17] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-Vocabulary Point-Cloud Object Detection Without 3D Annotation

[18] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences.

.

.

.