Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag

A benchmark approach and dataset for large-scale lane mapping from MLS point clouds

Xiaoxin Mi [a,b,d], Zhen Dong [b], Zhipeng Cao [c], Bisheng Yang [b], Zhen Cao [b], Chao Zheng [c,*], Jantien Stoter [d], Liangliang Nan [d]

[a] *School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, Hubei, China*
[b] *State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Luoyu Road 129, Wuhan, 430079, Hubei, China*
[c] *Center of Data Intelligence, Tencent, Beijing, China*
[d] *Urban Data Science, Delft University of Technology, Julianalaan 134, Delft, 2628 BL, The Netherlands*

ARTICLE INFO

Keywords:
Large-scale lane mapping
Point clouds
End-to-end
Neural network
Hierarchical attention

ABSTRACT

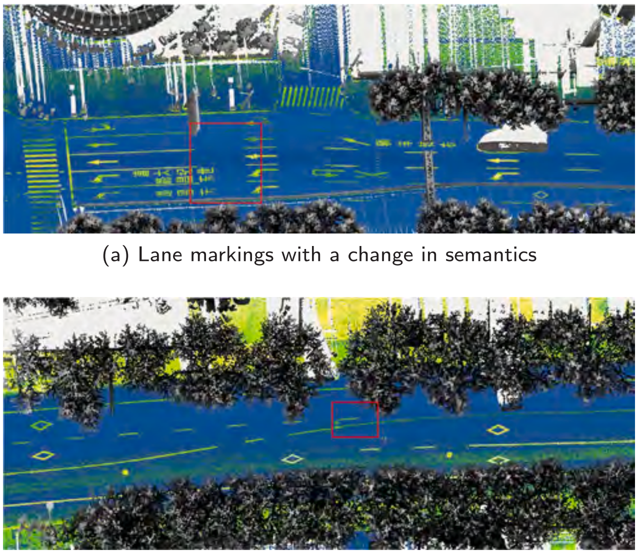
Accurate lane maps with semantics are crucial for various applications, such as high-definition maps (HD Maps), intelligent transportation systems (ITS), and digital twins. Manual annotation of lanes is labor-intensive and costly, prompting researchers to explore automatic lane extraction methods. This paper presents an end-to-end large-scale lane mapping method that considers both lane geometry and semantics. This study represents lane markings as polylines with uniformly sampled points and associated semantics, allowing for adaptation to varying lane shapes. Additionally, we propose an end-to-end network to extract lane polylines from mobile laser scanning (MLS) data, enabling the inference of vectorized lane instances without complex post-processing. The network consists of three components: a feature encoder, a column proposal generator, and a lane information decoder. The feature encoder encodes textual and structural information of lane markings to enhance the method's robustness to data imperfections, such as varying lane intensity, uneven point density, and occlusion-induced incomplete data. The column proposal generator generates regions of interest for the subsequent decoder. Leveraging the embedded multi-scale features from the feature encoder, the lane decoder effectively predicts lane polylines and their associated semantics without requiring step-by-step conditional inference. Comprehensive experiments conducted on three lane datasets have demonstrated the performance of the proposed method, even in the presence of incomplete data and complex lane topology. Furthermore, the datasets used in this work, including source ground points, generated bird's eye view (BEV) images, and annotations, will be publicly available with the publication of the paper. The code and dataset will be accessible through here.

1. Introduction

Lane markings on road surfaces are typically painted as continuous curves or dashed lines using high retro-reflective materials, as depicted in Fig. 1. These markings define the driving-free area for vehicles and play a crucial role in ensuring safe, efficient, and comfortable driving. Lane information is essential for vehicle intelligence, as it provides vital prior knowledge for tasks such as sensing surroundings, planning future routes, and making correct decisions and reactions. Additionally, accurate lane mapping is vital to create digital twin cities that aim to replicate the real world, including detailed lanes. Moreover, the semantics of lane markings indicates the connectivity between adjacent lanes, in which dashed lines indicate connectivity, while continuous curves indicate isolation and non-changeable lanes. Traditional lane mapping procedures typically involve several steps (Mi et al., 2021a). Initially, data is collected by vehicles equipped with mapping sensors and traversing roads in the target mapping area. Subsequently, the collected source data is geo-referenced and reorganized. Finally, artificial interpretation is performed, where annotators manually draw lane lines to generate a vectorized representation. However, this manual annotation process is time-consuming and labor-intensive. Therefore, researchers have been exploring automatic lane extraction methods

[*] Corresponding author.
E-mail addresses: xiaoxin.mi@whut.edu.cn (X. Mi), dongzhenwhu@whu.edu.cn (Z. Dong), galaxycao@tencent.com (Z. Cao), bshyang@whu.edu.cn (B. Yang), zhen.cao@whu.edu.cn (Z. Cao), chrisczheng@tencent.com (C. Zheng), j.e.stoter@tudelft.nl (J. Stoter), liangliang.nan@tudelft.nl (L. Nan).
URL: https://3d.bk.tudelft.nl/liangliang/ (L. Nan).

https://doi.org/10.1016/j.jag.2024.104139
Received 8 December 2023; Received in revised form 22 July 2024; Accepted 1 September 2024



(a) Lane markings with a change in semantics

(b) A change in lane number

Fig. 1. Typical urban road scenarios captured from MLS point clouds. The distribution of lane markings in urban road scenarios exhibits significant variations, including changes in lane number and semantics. (a) Illustration of lane markings with a change in semantics, where a dashed lane marking transits to a continuous lane marking. (b) Illustration of a change in lane number due to lane forking. The ground points are visually represented with descending intensity, ranging from yellow to blue, while the non-ground points are rendered in grayscale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that utilize various data sources, primarily optical images and light detection and ranging (LiDAR) point clouds.

The related methods can be divided into two categories according to application scenarios: real-time and offline lane detection. Real-time lane detection requires efficient and low-computational cost algorithms to support real-time applications such as autonomous driving. The source data is usually optical images, which lack geo-referenced elevation and usually suffer from distortion and degradation. As a result, the detected results are typically unsuitable for generating high-definition maps (HD Maps). In contrast, point clouds collected by mobile laser scanning systems (MLS) provide accurate three-dimensional geometry and other attributes, such as intensity, on the discrete surfaces of objects, making them a kind of common source data for digital twins (Soilán et al., 2021). Unlike real-time lane detection methods, offline lane detection methods prioritize generating accurate and continuous lane maps and are less concerned with computational efficiency. This paper focuses on offline lane marking detection and proposes a method to generate accurate and reliable lane maps using MLS point clouds.

Many lane detection methods assume that lane markings have slender shapes and utilize the contrast in intensity values or color differences between lane markings and the background in point clouds or optical images, respectively. Consequently, these methods can be categorized into unstructured and structured lane detection approaches. Unstructured lane detection methods primarily rely on intensity or color difference to extract lane markings, while structured lane detection methods additionally consider shape characteristics. For instance, typical unstructured lane detection methods, such as segmentation-based methods, first separate dense lane marking points from the background, then cluster and fit the detected points to generate vectorized lane markings. The final performance heavily relies on the segmentation results, which are easily affected by varying intensity distribution and incomplete data (Yang et al., 2012; Yu et al., 2014; Wen et al., 2019; Cao et al., 2023). To generate vectorized lane markings straightforwardly and efficiently, researchers propose structured lane detection methods, including ordered points-based methods (Paek et al., 2022; Qin et al., 2022) and parametric model-based methods (Neven et al., 2018; Feng et al., 2022). Ordered points-based methods detect sparse ordered points without sequential segmentation and clustering procedures. In these methods, effectively encoding the structural information between and within lanes is crucial to enable simultaneous sensing of multiple lanes. On the other hand, parametric model-based methods utilize lines, splines, or Bezier curves to regress parametric lane boundaries. While these methods have brief and concise pipelines, the accuracy of the regressed lane boundaries is reported to be lower than the ordered points-based methods, as noted by Liu et al. (2021a).

Since the remarkable success of transformers based on the attention mechanism in sequence tasks (Vaswani et al., 2017; Dosovitskiy et al., 2020) applied a transformer called vision transformer (ViT) directly to sequences of image patches for image classification. Experiments have demonstrated that ViT, pre-trained on large-size datasets and transferred to a smaller dataset, can attain good results compared to the state-of-the-art CNN models while requiring fewer computational resources to train. Subsequently, numerous transformer-based works came out. Researchers apply vision transformer and its variants (Liu et al., 2021b) to object detection (Carion et al., 2020; Li et al., 2022a), semantic segmentation (Xie et al., 2021; Strudel et al., 2021), and instance segmentation, which perform excellently as well in global feature learning and overall performance.

Based on the aforementioned discussion, the primary challenges in large-scale lane mapping are outlined as follows. Firstly, the inclusion of both geometric and semantic information is crucial for lane mapping and its associated applications, such as route planning and navigation. However, existing methods predominantly concentrate on lane geometry while disregarding semantic aspects. Secondly, conventional object detection and segmentation networks in computer vision are not suitable for lane marking mapping due to their elongated and slender shape. Consequently, it is imperative to consider both local patterns and global shapes in automatic lane mapping approaches. Lastly, the lack of dense point cloud datasets for large-scale lane mapping limits the relevant research.

This paper proposes a method to generate large-scale lane maps with associated semantics. The main contributions of the proposed method are as follows:

- We propose an end-to-end lane mapping approach incorporating lane markings' textural, shape, and structural characteristics. This method allows for the simultaneous mapping of lane markings and their corresponding semantic information, eliminating the need for intricate postprocessing. Besides, our proposed method exhibits strong generalization capabilities, particularly in demanding urban environments with fragmented data and intricate road topologies.
- We introduce a hierarchical lane attention mechanism to capture both intra-lane and inter-lane structures using the vision transformer at a global scale and spatial attention inside the local column proposal. This module enhances the network's robustness against incomplete data and improves the accuracy of the detected lanes.
- To facilitate future research in the field, we will release a lane dataset with comprehensive semantics, including dashed and continuous lane markings. This dataset encompasses a road length of over 98 km and comprises 102 GB of point clouds. To enhance processing efficiency, we have organized the point clouds into segments. Each road segment measures approximately 50 m in length and 25 m in width, aligned with and perpendicular to the direction of the road. Furthermore, each segment is accompanied by ground point clouds, corresponding Bird's Eye View (BEV) images, and annotations. Notably, all lanes are labeled at the instance level. It is worth highlighting that this dataset is the first dense MLS point cloud dataset captured in real-world scenarios for large-scale lane mapping.

The remainder of the paper is organized as follows. Section 2 reviews and discusses the relevant studies on lane detection and delineation. Section 3 elaborates on the proposed approach. Section 4 presents the datasets, implementations, experimental results, comparisons, and ablation studies. The conclusion is outlined in Section 5.

2. Related works

Lane markings can be detected using textual and shape information. According to the representation of the detected lanes, related studies can be categorized into two groups: unstructured lane detection and structured lane modeling (Tang et al., 2021). Unstructured lane detection methods distinguish lane markings by color or intensity difference from the background. Depending on the source data, the detected lane markings are clusters of pixels or points. On the other hand, structured lane modeling methods consider the geometric shape of lanes and represent them as geometric line-type models.

2.1. Unstructured lane detection

Traditional unstructured lane detection methods (Bar Hillel et al., 2014) rely on the difference in visual appearance between lane markings and road surfaces. These methods typically utilize exact colors, simple gradients, and brighter intensity to differentiate lane markings. For instance, in Choi et al. (2022), lane markings are filtered using an intensity threshold. Similarly, in Lin et al. (2021), lane markings are automatically extracted from the sparse LiDAR point clouds using intensity threshold at first, then clustered using an adaptive clustering method. Although these methods can recognize lane markings based on low-level features without needing labeled data or trained models, their performance degrades in complex scenarios such as lanes in shadow, intense lighting, and incomplete data. Furthermore, it is impossible to simultaneously predict the semantics (continuous or dashed) of lane markings.

With the significant advancements in deep neural networks, segmentation-based methods have been introduced in lane detection tasks (Tang et al., 2021). In Zang et al. (2017), lanes were extracted from satellite imagery using a hierarchical segmentation strategy. The authors first identified cropped samples with road markings and then calculated pixel- and sub-pixel-level lane locations via a Gaussian model. Candidate lines were grouped, classified, and interpolated to obtain complete, functional, and structured lane lines. Another approach proposed by Li et al. (2021) is HDMapNet, which generates a locally consistent map for real-time motion planning. It involves feature encoding (fusion of optical images and point clouds into feature maps) and prediction decoding (predicting semantics, instances, and directions). However, the final vectorized results were clustered and skeletonized based on the three predictions in the postprocessing step. Similarly, in Li et al. (2022b), lanes were extracted from the project BEV images using an existing instance segmentation network. In addition to lane detection, some studies have focused on lane connectivity at intersections. Tardy et al. (2023) and Soilán et al. (2022) segment the road marking points from the MLS point clouds using a Point Transformer model. Following the point cloud semantic segmentation, road marking points and MMS (Mobile Mapping System)'s trajectory are used to extract road centerlines and other road inventory parameters. Zhou et al. (2021) proposed a semantic particle filter-based approach to construct the lane centerline maps, where the feature of particles was embedded using a segmentation network. Road topological connections at intersections were inferred based on the constructed lane centerlines and OSM (Open Street Maps) road skeletons. Similarly, He and Balakrishnan (2022) extracted the lane centerline and its direction from patched aerial images using a semantic segmentation network and subsequently established the connectivity among terminal vertices at intersections.

Thanks to the improved generalization performance of neural networks, lane segmentation methods based on neural networks exhibit greater robustness than traditional threshold-based methods. However, these methods still face challenges in handling occlusions, which may result in incomplete lane extraction and hinder lane structural encoding. Additionally, the segmented dense lane pixels or points generated by these methods are unordered, redundant, and lack structure. Therefore, postprocessing steps are still required for subsequent applications. This work proposes an end-to-end structured lane mapping method robust to imperfect data to address these issues.

2.2. Structured lane modeling

Compared to unstructured lane extraction methods, end-to-end structured lane modeling methods can generate concise lane representations that can be directly utilized or require only a few postprocessing steps for subsequent applications (Hervieu et al., 2015). There are two types of structured lane modeling methods according to the underlying lane representations: ordered points-based and parametric model-based approaches.

Ordered points-based methods formulate the lane as a set of sparse ordered points. Several methods adopted RNN (Recurrent Neural Network) to trace the Mordered points. For instance, Homayounfar et al. (2018) developed a hierarchical recurrent network to generate lane polylines. The process began by detecting the initial region for each lane boundary and then using RNN to trace all connected lane regions. However, the resulting region-level lanes are often too coarse for generating precise lane maps. Further postprocessing steps, such as threshold segmentation, clustering, and skeletonization, were employed to obtain accurate lane marking polylines. To improve accuracy, efficiency, and automation, Homayounfar et al. (2019) created lane maps of highways using a directed acyclic graphical model. By encoding the geometrical and topological properties of the local region of the lane boundary in graph nodes, lane boundaries can be extracted by optimizing the graph nodes on a given BEV image and finding a maximum posterior over the space of all possible graphs. This model struggles with missing data, a common issue in data acquisition due to unavoidable occlusions. Furthermore, in RNN-based methods, earlier vertices conditionally contribute to later vertices, leading to inaccuracies and limiting computational efficiency (Tian et al., 2022). To address these challenges, line anchors were introduced in lane detection tasks, taking inspiration from object detection techniques and the slender shape of lanes. Garnett et al. (2019) processed a single image in two parallel streams: the raw image-view pathway and the top-view pathway. This allowed for the simultaneous extraction of lane-related features. The vertices were then regressed using equally spaced longitudinal anchors to represent lanes. Because of the longitudinal assumption of lane anchors, this method has difficulty handling lanes with complex topologies, such as urban intersections.

For lanes with complex topologies, Efrat et al. (2020) enhanced the 3D LaneNet (Garnett et al., 2019) to create 3D LaneNet+. This improved model learned local feature encoding and global connectivity embedding to generate complete 3D lanes. The global lanes can then be grown from the local parametric lane segments. In other approaches, transformers were utilized to capture global features and directly regress row-wise ordered points. Tabelini et al. (2021) used line anchors for local lane feature pooling and aggregating global lane attention. Based on the aggregated feature maps, anchor probability, horizontal offsets of vertices, and lane length were estimated. Similar to previous methods, it did not explicitly consider the lane's topology. To address complex topologies, CondLaneNet (Liu et al., 2021a) introduced the RIM (Recurrent Instance Module), which can differentiate overlapping lines and deduce row-wise vertices for each lane instance. Another approach proposed by Paek et al. (2022) was a Lidar-based lane detection network that utilized the self-attention mechanism to extract global correlations. This model outperformed the lane detection models by only using corresponding CNN for feature encoding, but it did not consider lane semantics and topology. Qin et al. (2020) proposed a row anchor-based selecting method on the global feature map method to detect lanes. However, the row anchor system may result in magnified localization errors when the lane direction is nearly parallel to the row anchor. To solve this issue, hybrid anchors were proposed to reduce the magnified localization error (Qin et al., 2022).

As for the parametric model-based methods, Hough Transform (Zheng et al., 2018), RANSAC (Borkar et al., 2009), and polynomial fitting (Neven et al., 2018) are conventional techniques that fit parametric models from segmented lane pixels or points. In addition to classic techniques, some methods directly regress curve models according to the observed geometrical shape of the lane. Neural networks have also been employed to estimate curve coefficients in these approaches. In Jin et al. (2022), an eigenlane space was constructed through low-rank approximation. Lane boundaries were then detected using an anchor-based network within this eigenlane space. The eigenlane space was created by analyzing the lane distribution in the training dataset and constructing lane bases accordingly. Methods such as NMS (Non-Maximum Suppression), inter-lane correlation, and maximum weight clique selection were employed to reduce lane redundancy and refine lane geometry. Another method proposed by Feng et al. (2022) utilized parametric Bezier curves to fit lanes in RGB images. This approach incorporated the horizontally flipped version of the feature map and applied sampled L1-loss to improve performance. However, it is essential to note that the symmetrical assumption of lane distribution is applicable only in limited scenarios, and further feature augmentation methods are still needed to enhance the performance of these techniques.

In summary, unstructured lane detection methods focus on utilizing local textural features of lanes but overlook lane structures within the segmentation network. On the other hand, structured lane modeling methods consider textual and structural information, making them more prevalent in recent research. However, RNN-based methods may struggle to predict continuous and complete lanes when faced with incomplete data. Additionally, most existing methods focus on lane geometry detection while neglecting lane semantics, which is crucial for lane maps and should be predicted simultaneously. This paper proposes a structure-aware lane mapping method that effectively encodes local textual features and global structural information about lanes. Our approach is robust against imperfect data, such as worn lane markings and incomplete data, while also handling lanes' topological and semantic changes in complex urban scenarios.

3. Methodology

We aim to generate 3D lane maps from MLS point clouds automatically. To enhance efficiency, we adopt a progressive approach by growing the lane map from local to large scale and reorganize point clouds as described in Mi et al. (2021b). The process involves partitioning point clouds along the trajectory, filtering out points above the corresponding trajectory, and generating BEV images. Since lanes typically exhibit a slender shape along the trajectory direction, local features with limited receptive fields may not fully capture the entire lane instance, especially for dashed and incomplete lanes. To address this, we formulate lane map generation as a top-down polyline detection and regression problem rather than extracting lanes in a bottom-top manner. To generate accurate lane maps with semantics (i.e., dashed or continuous), we propose a method that simultaneously estimates geometrical and semantic lane information, considering local textural features and global structures. In the following sections, we describe the problem formulation and present the three components of our network: feature encoder, proposal generator, and lane decoder.

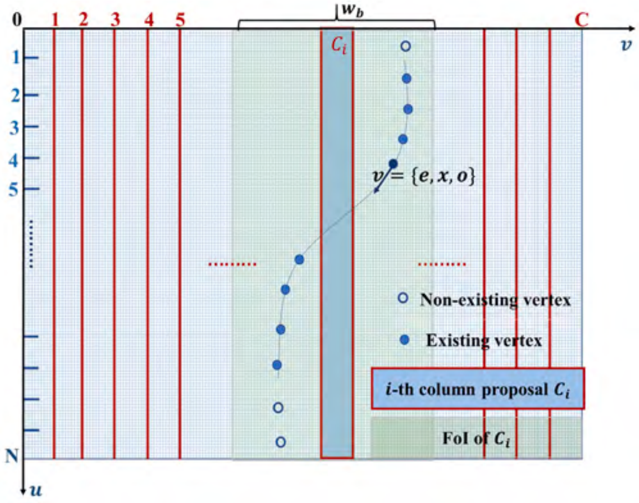


Fig. 2. Schematic diagram of the column proposal-based formulation for lane markings. C initial column proposals are displayed in the horizontal direction. Each polyline in the FoI of a proposal has a buffer width of w_b and a maximum of N sampled points in the vertical direction. The existence e, coordinate x, and orientation o of each vertex v are to be estimated.

3.1. Problem formulation

Our approach is inspired by the distribution of lane markings on road surfaces, where they are multiple polylines painted with specific textural patterns (continuous curves or dashed lines corresponding to their semantics). Based on this observation, we can effectively represent the lane geometry as a polyline consisting of a sequence of ordered vertices. Let $P = \{V_i\}, i = 1, 2, \ldots, N$ be a lane polyline, each vertex V_i contains information about the existence e_i, coordinates x_i, and orientation o_i of the polyline as $V_i = \{e_i, x_i, o_i\}$. These components together determine the geometry of the lane: existences $\{e_i\}$ determine the semantic existence probability, coordinates $\{x_i\}$ specify the accurate location, and orientations $\{o_i\}$ indicate the stretching direction of the polyline at the exact location, respectively, as shown in Fig. 2.

With the assumption that lane markings are represented as polylines, we propose our one-stage lane mapping network, which can estimate multiple lanes with associated semantics end-to-end. The detailed network architecture is described in the next section.

3.2. BEV projection

To address the task of large-scale lane mapping, we employ a method wherein the source point clouds are systematically divided into patches. Local lane maps are then modeled within these individual patches, subsequently contributing to the development of expansive lane maps on a larger scale. Specifically, point clouds are cropped into patches, and each patch has a length of 50 m along the road stretching direction and a width of 22 m perpendicular to the trajectory direction. Then, point clouds in each patch are rotated along the z-axis to set the stretch of the road as the y-axis, as shown in Fig. 3. 3D points below the vehicle's trajectory are projected onto horizontal planes to form BEV images with a fixed resolution.

To facilitate lane mapping, we encode specific attributes of point clouds and the corresponding vehicle trajectories into BEV images, including the mean intensity inside the pixel, the distance to the nearest trajectory of the pixel, and the lowest elevation within each pixel, which correspond to B, G, and R channels, respectively. Because the intensity difference between the lane markings and the surrounding road surface points serves as a discriminating feature for lane markings recognition in point clouds. Moreover, as intensity tends to decrease

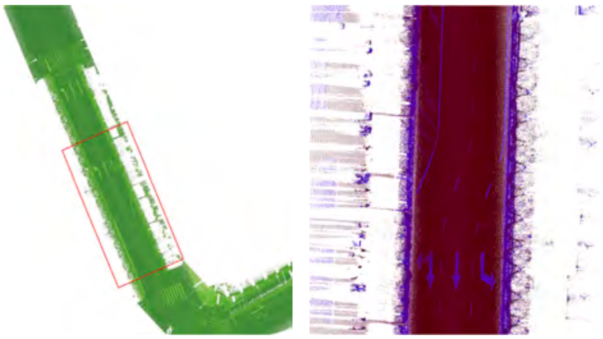

(a) Input point cloud (b) Projected BEV image

Fig. 3. BEV image generation from the source MLS point cloud. (a) displays part of the input point cloud, rendered with descending intensity from yellow to green. The region marked by the red rectangle is cropped to generate the BEV image shown in (b). The BEV image encodes intensity, local minimum elevation, and nearest distance to the trajectory in its B, G, and R channels, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with increasing scanning range, we encode the nearest distance to the trajectory in the BEV image, following a similar approach in previous work (Mi et al., 2021b). Additionally, the lowest elevation within each pixel is encoded in the BEV image to aid in the recovery of detected lane polylines from the BEV image to the source point cloud coordinate system. Fig. 3 illustrates the BEV image generation procedure.

3.3. Feature encoder

Fig. 4 illustrates the architecture of the proposed column proposal-based network for end-to-end lane map generation. Like other methods, we employ the convolution neural network to embed the input BEV images into feature maps $\mathscr{F}_1$. Following this, we use a vision transformer (ViT) backbone to sense the global relations of lanes since lane markings are typically slender and extend across the entire image. During the procedure, $\mathscr{F}_1$ is then split into several patches and, after positional embedding, fed into the ViT encoder to sense the global features $\mathscr{F}_2$. At the same time, we use the feature pyramid network to aggregate the hierarchical features $\mathscr{F}_3$ to preserve lane details. $\mathscr{F}_2$ and $\mathscr{F}_3$ are concatenated into $\mathscr{F}_4$, which consists of both local and global features to support lane mapping.

3.4. Column proposal generator

After the BEV projection (see Section 3.2 and Fig. 3), the lanes in BEV images are aligned vertically with a uniform scale. Therefore, we use natural column proposals on a sole-scale feature map to predict lane polylines, which differs from the object detection network FCOS (Tian et al., 2019) predicting objects on hierarchical pyramid feature maps. To generate the column proposal, we crop the feature map $\mathscr{F}_4$ for each column proposal C_i with a fixed buffer area as its feature of interest (FoI), as shown in Fig. 2. The extended feature area aims to increase the receptive field of the column proposal and strike a balance between continuity and redundancy. Unlike the anchor-based methods that introduce additional anchor parameters or RoI alignment (Ren et al., 2015), we directly detect and regress lanes as polylines inside the FoI after column proposal generation. The learning target T_{C_i} of column proposal C_i is the closest ground truth lane marking to the center of FoI, which can be calculated as

$$T_{C_i} = \min_{l=1}^{K} \left\{ \frac{\sum_{j=1}^{N} ext_{i,l}^{j} |C_i^j - T_l^j|}{1 + \sum_{j=1}^{N} ext_{i,l}^{j}} \right\}, \tag{1}$$

$$ext_{i,l}^{j} = \begin{cases} 1, & |C_i^j - T_l^j| < \frac{w_b}{2} \quad \& \quad T_l^j > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where K denotes the number of ground truth lanes. N is the maximum sample points number of a lane polyline. $ext_{i,l}^{j}$ represents the existence of jth vertex of the ground truth lane l in the ith column proposal C_i. C_i^j is the column coordinates of the jth vertex of column proposal C_i. T_l^j is the column coordinates of the jth vertex of ground truth T_l, and w_b is the buffer width. Proposals with more than two sampled points inside are considered positive samples, while those with fewer than two sampled points are negative.

For each column proposal, we estimate its objectiveness, which indicates the confidence of the lane within the proposal. During training, the Binary Cross-Entropy loss function is adopted to supervise this branch, with a l_{obj} loss. For inference, the predicted objectiveness confidence is utilized to filter out polylines in low-confidence proposals.

Column spatial attention. To achieve promising performance in lane detection, we have designed a segmentation-driven spatial attention mechanism operated as follows. We initially estimate binary segmentation within each column proposal. Subsequently, we multiply the binary segmentation feature explicitly with the FoI of the corresponding column proposal in a location-wise manner, thereby augmenting the subsequent lane decoder. This design confines spatial attention within the column proposal and ultimately improves the overall lane mapping accuracy. The auxiliary semantic segmentation inside the FoI of the column proposal is intended to enhance the overall performance. To supervise the semantic branch, the binary cross-entropy loss l_{sem} is employed as the loss function.

3.5. Lane marking decoder

As explained in Section 3.1, the representation of a lane marking is a sequence of ordered points sampling on itself. The geometry heads within the network are responsible for estimating the existences, coordinates, and stretching directions of these ordered points denoted as $V_i = \{e_i, x_i, o_i\}$, within positive column proposals. The existence e_i of a vertex can be classified as *null*, *dashed*, or *continuous*. The *null* state indicates the absence of a lane polyline at the given location, while the *dashed* and *continuous* states signify the presence of the lane and its associated semantics. The coordinate x_i encompasses each vertex's integer value and additional offset. As lane marking points are uniformly sampled along the vertical axis on the BEV images, it is easy to calculate points' y-values in the local coordinate system. Consequently, only x-values of the points are required for prediction. To transform coordinate regression into a classification problem, we discretize the search space into X consecutive categories. We employ a similar approach to the stretching direction o_i by uniformly dividing the stretching space $[0, \pi]$ into B bins. The three geometry-related heads for points in the identical column proposal share the foundational geometrical layers and independently predict their respective parameters within the network.

The existence and stretching direction loss functions are formulated as the cross-entropy loss. As for the coordinate estimation head, we employ a combination of a cross-entropy loss and a smooth-l_1 loss. Cross-entropy loss is adopted to supervise the integer part of x-values. In addition, we draw inspiration from Liu et al. (2021a) and calculate the smooth-l_1 loss using expectations of predicted x-values of the vertex to avoid discontinuous derivatives at extreme values. Notably, only the offsets of the nearest 5 locations are considered in calculating the coordinates' offset loss by smooth-l_1 loss. The overall geometrical loss is given by the weighted sum of the above losses as

$$l_{geo} = [w_e, w_{x_c}, w_{x_s}, w_{x_o}, w_o]^T [l_e, l_{x_c}, l_{x_s}, l_{x_o}, l_o], \tag{3}$$

where l_e, l_{x_c}, l_{x_s}, l_{x_o}, and l_o are the existence loss, the coordinate cross-entropy loss, the coordinate smooth l_1 loss, the coordinate offset loss, and the stretching direction loss, respectively. The weights of the corresponding losses are the $\{w_i\}$.

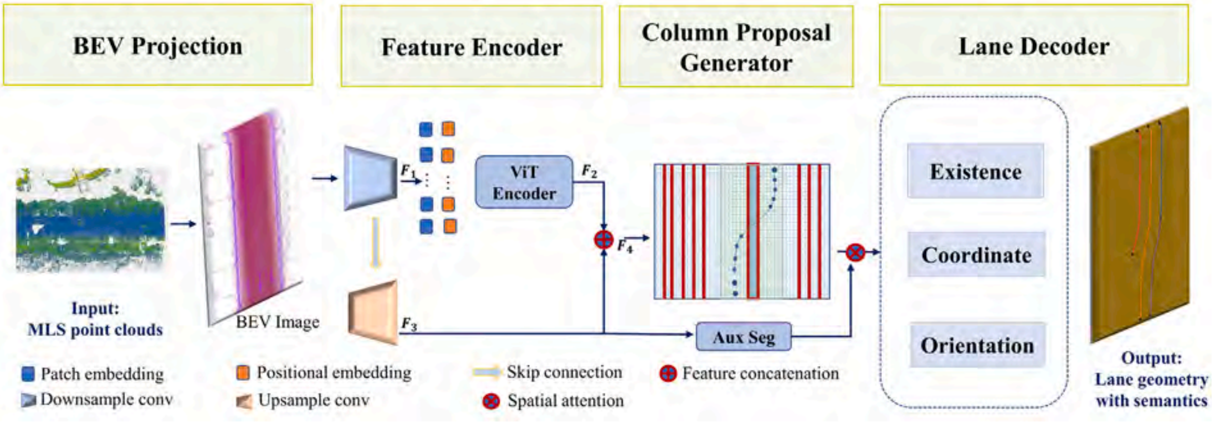


Fig. 4. Network architecture: proposed column proposal-based model for lane mapping. The network is applied to the projected LiDAR BEV images and outputs the lane marking geometry as its centerline with the associated semantics at the instance level.

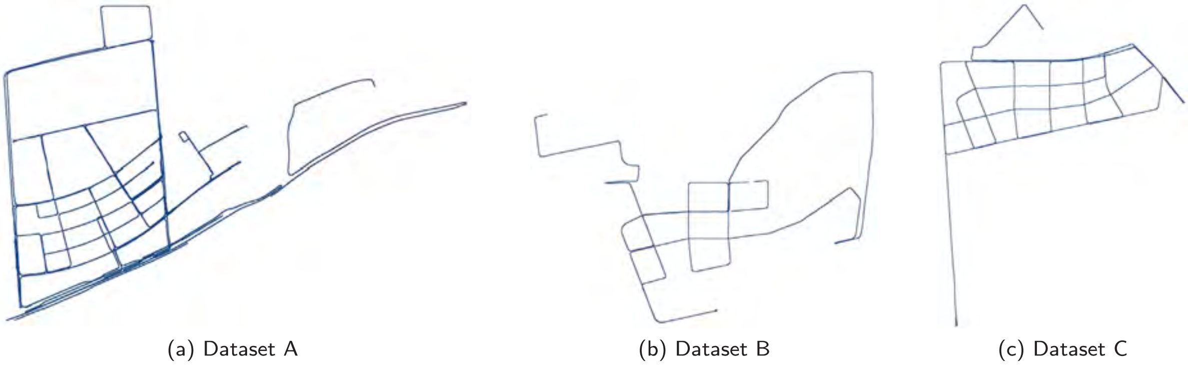


Fig. 5. Vehicle trajectories of the three datasets.

3.6. Implementation details

Training loss. With the above-defined objectiveness loss l_{obj}, semantics loss l_{sem}, and geometry loss l_{geo}, our overall training loss function can be the sum of the three loss terms,

$$loss = l_{obj} + l_{sem} + l_{geo}, \tag{4}$$

Inference. In the inference stage, accurate and concise lane maps are generated by considering several factors, including the proposal objectiveness confidence, the existence confidence of sampled points, the regressed coordinates, and the predicted stretching directions. Specifically, the proposal objectiveness confidence filters out lane polylines predicted from column proposals with low confidence. Subsequently, the predicted ordered vertices with satisfactory existence confidence scores are connected to form the initial lane polylines. A line NMS algorithm is adopted to mitigate redundancy among the regressed lane polylines from all column proposals, and it effectively selects the most representative and relevant lane polylines.

Lane map expansion. With the success of predicting local lane maps on BEV images, we back-project the predicted lane polylines via the reverse geometric operation of BEV generation from the local BEV coordinate system to the source MLS point cloud system sequentially. Specifically, the stored lowest elevation in each pixel on BEV images is employed to recover the elevation value of lane maps. During the expansion, lane polylines' semantics are attached for each sampled point as the associated attribute.

Table 1
Summary of each dataset's road length, data size, and scenario.

Dataset	Road length (km)	Data size (Source points) (GB)	Scenario
A	58.182	79.5	Urban area + expressway
B	17.243	28.2	Urban area
C	22.534	34.0	Urban + rural area

4. Experiments

4.1. Dataset description

The LiDAR point clouds utilized in our experiments were captured using a laser scanning system installed on a mobile surveying vehicle. The data collection occurred in Shanghai, China, encompassing three districts, corresponding to three datasets respectively. Specifically, District A includes urban and expressway scenarios, District B focuses on urban areas, and District C covers multiple urban areas and also a few rural regions. Each district's total length, data size, and scene scenario are summarized in Table 1. In addition to the two representative road surface scenarios of our datasets presented in Figs. 1, 5 shows the complete scanning routes in three districts, respectively.

Lane annotation. Ground truth annotation plays a crucial role in supervised learning. In this work, we enlisted experienced annotators to annotate three-dimensional lane markings using polylines with semantic labels for the three datasets. Guiding by the BEV projection with the resolution of 1152 by 1152, we generate the ground truth targets that include lane marking instance polylines, lane semantic polylines, and stretching directions. An example of the source ground

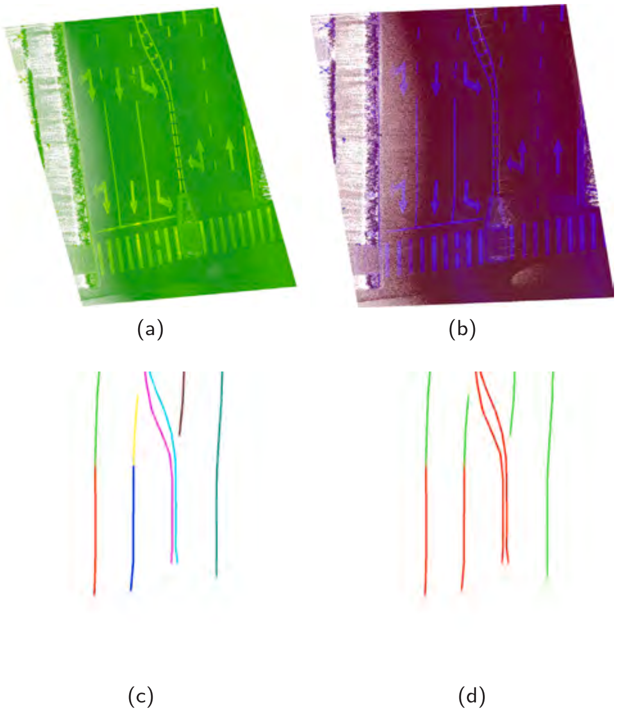


Fig. 6. An example of the source point cloud and the corresponding annotations. (a) Source point cloud rendered in descending intensity, from bright yellow to green. (b) Corresponding BEV image. (c) Corresponding lane instances. (d) corresponding lane semantics. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

point cloud and the corresponding annotations are illustrated in Fig. 6. To facilitate reproducibility and further research, all training data, including source ground points, projected BEV images, annotations, and generated ground truth targets, are made available alongside the publication of this paper.

4.2. Evaluation metrics

In line with the evaluation approach employed in previous works, such as DAGMapper (Homayounfar et al., 2019), we assess the performance of the proposed method on both BEV images and source point clouds using the precision, recall, and F_1 metrics. For the geometrical evaluation, we uniformly sample the ground truth and the predicted lane polylines into sets of ordered vertices. True Positives (TP) are identified when predicted vertices fall within the buffer zone of ground truth vertices. False Positives (FP) occur when predicted vertices do not intersect with any ground truth vertex buffer zone, while False Negatives (FN) correspond to ground truth vertices without any predicted vertices falling within their buffer zone. By tallying the numbers of samples in TP, FP, and FN, we calculate the precision, recall, and F_1 metrics. Additionally, by multiplying the sample interval by the respective counts of TP, FP, and FP at different buffer ranges, the overall length of the lane can be estimated. Semantic accuracy is evaluated using a similar approach but with the inclusion of semantic constraints. Specifically, semantic True Positives (TP) are vertices falling within the buffer zone of the ground-truth vertices, when predicted vertices and ground-truth vertices are with the same semantics. The semantic FP and FN are determined similarly.

4.3. Implementation

Training details. Our network is trained and validated on samples in District A using an Nvidia 4090 GPU. Specifically, 85% of the samples in District A are for training, while the remaining 15% is used for validation. We apply data augmentation on the training dataset, involving increasing the number of scenarios with curved lanes by intentionally cropping redundant ground blocks with curved lanes inside. ResNet-34 (He et al., 2016) is used for BEV image feature encoding in the feature encoder. We fine-tuned the model pre-trained on the ImageNet dataset (Russakovsky et al., 2015). Then, a vision transformer encoder is employed to capture global feature correlations among split feature patches. After feature encoding, the column proposal generator generates the column proposals. Subsequently, the lane decoder is applied for each column proposal. X consecutive category in coordinate estimation is set as the width of the column proposal's FoI. The number of bins B of stretching space is set as 9 in the experiment. The network parameters are learned with the Adam solver (Kingma and Ba, 2014) for 7K iterations with an initial learning rate of 1.5e−4 and a batch size of 6 images. Other parameters are set to their default values.

Inference details. During the inference process, an input BEV image is forwarded through the feature encoder, followed by column proposal generation and the prediction of proposal objectiveness confidences, geometrical existences, coordinates, and stretching orientations for each vertex in the proposal. The proposal objectiveness confidence is utilized to filter out polylines in low-confidence proposals, which is set as 0.2 after extensive experiments. For the proposal with objectiveness confidence higher than the threshold, the initial lane polyline inside is obtained by connecting the predicted vertices with a higher existence confidence score, which is set as 0.3 after extensive experiments. After that, a simple polyline NMS is applied to refine the initial polylines to exclude redundancy. After detecting lanes from the BEV images, the predicted polylines are projected back to the point clouds by applying the inverse transformation for BEV image generation. The z values are retrieved by utilizing the minimum elevation value associated with the pixels of the BEV images.

4.4. Experimental results

Qualitative results. In Figs. 7 and 8, we present the lane mapping results for the entire datasets B and C, respectively. In Fig. 7, the overall lane mapping result for Dataset B is shown on the left, while four enlarged views are presented on the right. Similarly, the overall lane mapping result on Dataset C is presented in the top left corner of Fig. 8, with several local views showing the lane mapping results overlaid on the ground point clouds. All these results demonstrate the effectiveness and robustness of our proposed method and its generalization capabilities.

Figs. 9, 10, and 11 present a few lane mapping results from different parts of datasets B and C. In Fig. 9, straight dashed and continuous lanes are stretching along the trajectory, and some lane markings are incomplete in the source point clouds due to occlusion caused by vehicles. These typical but less challenging straight lanes are common in both urban scenarios and expressways, which can be successfully reconstructed by the proposed method in terms of both geometry and semantics. Moreover, intersections are common in urban scenarios, and the lane mapping result in Fig. 9(a) demonstrates that our method can also reliably reconstruct lane markings at the intersections. Fig. 10 showcases the reconstructed complete and continuous lane mapping results using the proposed method in the scenario where lane forks and source lane markings are incomplete due to occlusion. Fig. 11 demonstrates the robust performance of our approach in handling curved lanes, lane forks, and lane merges, accurately capturing their variations, and maintaining consistency in the mapping results.

Quantitative results. Quantitative evaluation results on the BEV images of datasets B and C are reported in Table 2, under three different buffer sizes 5, 10, and 20 pixels. Furthermore, to demonstrate the accuracy of lane mapping within the original point cloud coordinate system, Table 3 presents the quantitative evaluation results for datasets B and C using buffer sizes of 10, 20, and 30 cm.

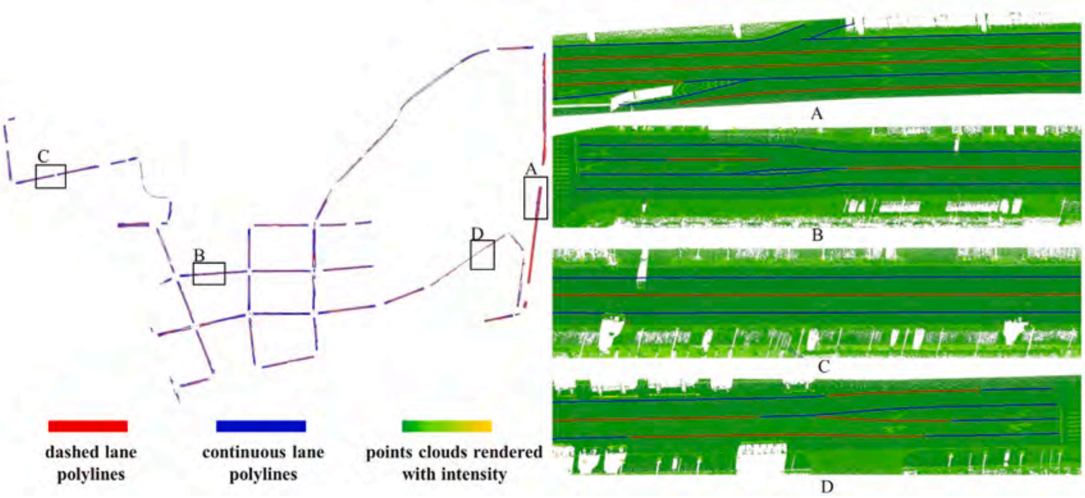

A
B
C
D
dashed lane polylines
continuous lane polylines
points clouds rendered with intensity

Fig. 7. Lane mapping results of dataset B.

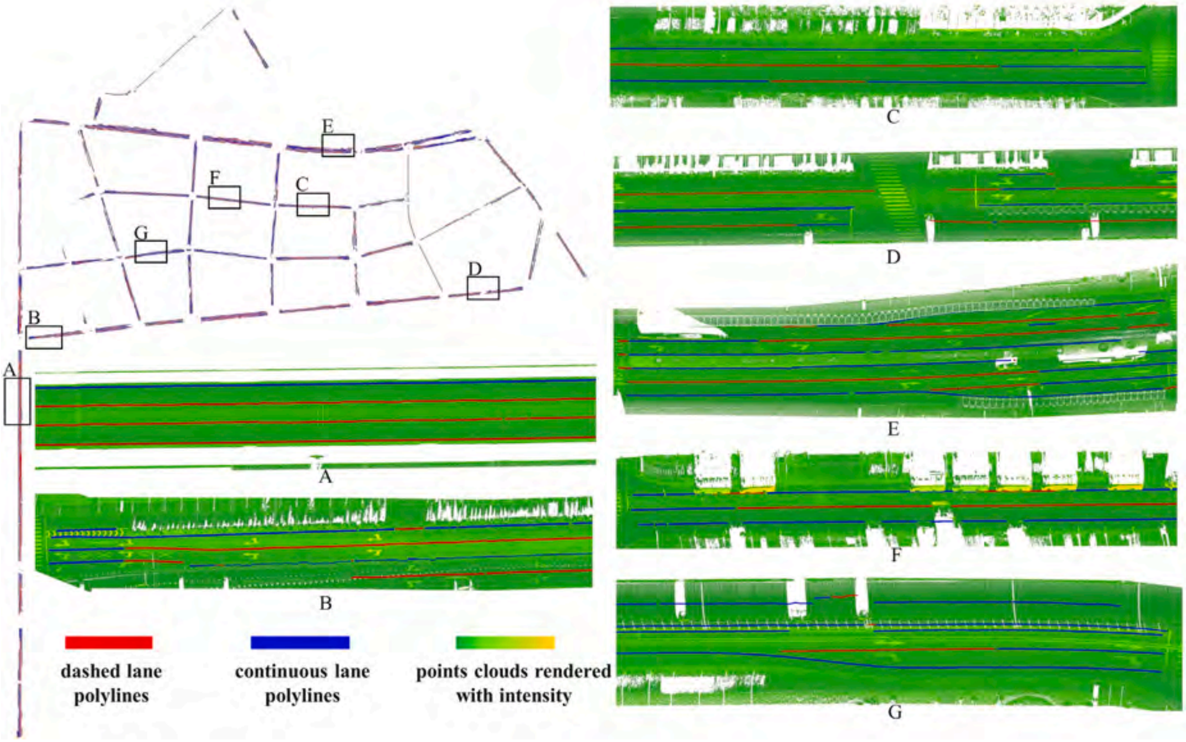

A
B
C
D
E
F
G
dashed lane polylines
continuous lane polylines
points clouds rendered with intensity

Fig. 8. Lane mapping results of Dataset C.

As can be seen from the statistical data presented in Table 2, our method demonstrates an overall precision and recall exceeding 0.88 with a buffer range of 20 pixels. Such performance enables the generation of expansive and intricate urban lane maps. It is noteworthy that the proposed method is anticipated to exhibit even higher efficacy in highway scenarios that demonstrate more distinct and linear lane markings and suffer from less data occlusion. The inference times for processing the complete datasets B and C amount to 214 s and 297 s, respectively.

Failure cases. We also observed a few failure cases in our experiments, as shown in Figs. 13 and 14. In Fig. 13, a few holes in the data lead to errors in semantic predictions. In Fig. 14(a) and (b), the road width exceeds the sensor's capacity to distinguish lane markings from the background, which results in incomplete results in the lane marking prediction. Furthermore, if two closely aligned lane markings are considered two separate lane markings than a whole object, the proposed method may perform poorly to accurately reconstruct two smooth parallel lane markings, as observed in Fig. 14(c). Though with

Table 2
Quantitative results on the datasets B and C, with a buffer range of 5, 10, and 20 pixels on the BEV images. The precision, recall, and F_1 metrics are reported.

Dataset		Buffer (5 pixel)			Buffer (10 pixel)			Buffer (20 pixel)		
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
B	Geometry	0.832	0.852	0.842	0.876	0.898	0.887	0.883	0.917	0.900
	Semantics	0.787	0.801	0.794	0.833	0.848	0.841	0.842	0.870	0.856
C	Geometry	0.815	0.768	0.791	0.885	0.835	0.859	0.903	0.873	0.888
	Semantics	0.748	0.706	0.726	0.831	0.785	0.807	0.854	0.821	0.837

Table 3
Quantitative evaluation of lane geometry on datasets B and C, with a buffer range of 10, 20, and 30 cm on the point clouds. TP, FP, and FN refer to lane length of the truly predicted, falsely predicted, and unpredicted lanes, respectively.

Dataset	Buffer (10 cm)						Buffer (20 cm)						Buffer (30 cm)					
	Precision	Recall	F_1	TP (km)	FP (km)	FN (km)	Precision	Recall	F_1	TP (km)	FP (km)	FN (km)	Precision	Recall	F_1	TP (km)	FP (km)	FN (km)
B	0.827	0.829	0.828	33.692	7.044	6.917	0.876	0.897	0.877	35.676	5.060	4.934	0.890	0.893	0.892	36.274	4.462	4.335
C	0.799	0.766	0.782	50.688	12.964	15.461	0.870	0.835	0.852	55.203	8.250	10.947	0.895	0.858	0.876	56.759	6.693	9.390

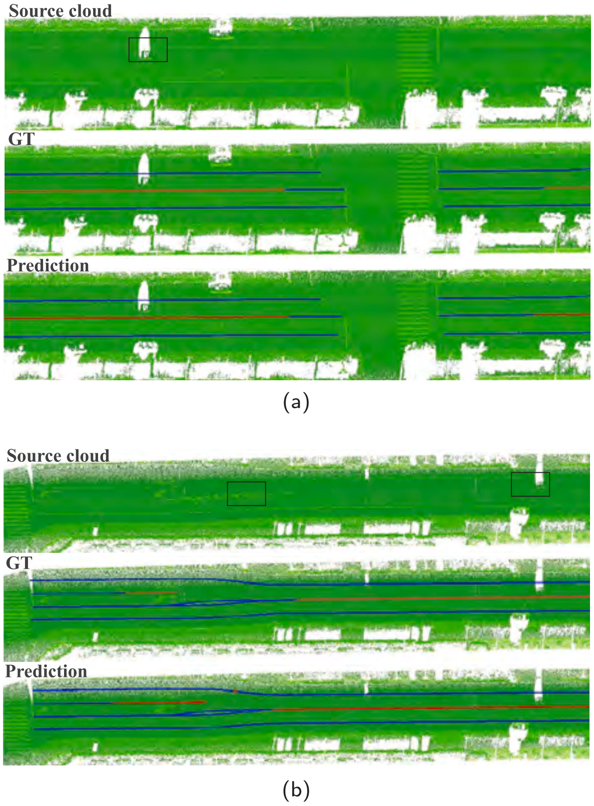


Fig. 9. Lane mapping results of two typical urban scenarios. (a) Lane mapping results on the road intersection. (b) Lane mapping results where source ground points are incomplete caused by occlusion. The ground point clouds are rendered with point intensity values. The dashed and continuous lane polylines are rendered in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

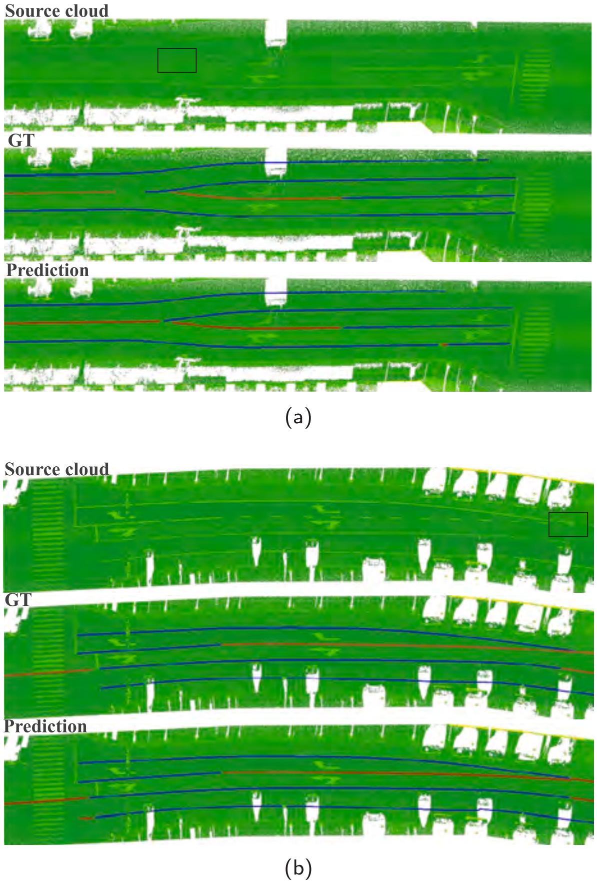


Fig. 10. Lane mapping results of a few typical urban scenarios where lane forks and source lane markings are incomplete due to occlusion. The ground point clouds are rendered with point intensity values. The dashed and continuous lane polylines are rendered in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

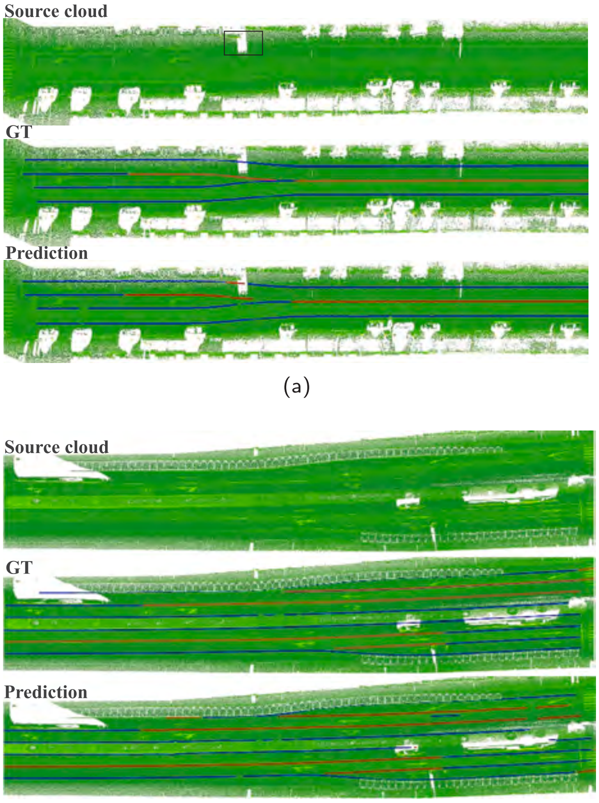
this failure case, it is worth noting that the method successfully maps two closely spaced parallel lane markings on the left side of Fig. 14(c). The failure is caused by the trade-off between the area of the column proposal's FoI and the polyline's continuity. Narrower FoI of column proposals have less potential to include more than one lane markings inside. However, it would make the predicted lane polylines less continuous. In our future work, parallel lane markings will be considered as another lane semantics with a parallel polyline pattern.

4.5. Comparison

We have also quantitatively compared our method with the state-of-the-art method KLane (Paek et al., 2022) and a classic FPN-based semantic segmentation method (Lin et al., 2017) on datasets B and C. The results are reported in Table 4. Our proposed method and KLane both utilize an ordered point sequence for lane representation. In the comparison, we employed the same network backbone for both

Table 4
Quantitative comparison of the proposed method with classic FPN-based semantic segmentation (Lin et al., 2017) and KLane (Paek et al., 2022) in terms of precision, recall, and F_1 metrics, with a buffer range of 5, 10, and 20 pixels on the BEV images. The ✓ indicates that a method is aware of the lane marking's Instance (Ins) or Semantics (Sem).

Method	Ins	Sem	Dataset B									Dataset C								
			Buffer (5 pixel)			Buffer (10 pixel)			Buffer (20 pixel)			Buffer (5 pixel)			Buffer (10 pixel)			Buffer (20 pixel)		
			Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Semantic segmentation		✓	0.438	0.546	0.486	0.524	0.662	0.585	0.578	0.736	0.648	0.323	0.372	0.346	0.442	0.519	0.477	0.514	0.622	0.563
KLane (Paek et al., 2022)	✓		0.553	0.516	0.534	0.728	0.652	0.688	0.774	0.690	0.730	0.517	0.485	0.501	0.692	0.619	0.654	0.758	0.688	0.721
Ours	✓	✓	**0.832**	**0.852**	**0.842**	**0.876**	**0.898**	**0.887**	**0.883**	**0.917**	**0.900**	**0.815**	**0.768**	**0.791**	**0.885**	**0.835**	**0.859**	**0.903**	**0.873**	**0.888**



(a)

(b)

Fig. 11. Lane mapping results in a few challenging scenarios: 11(a) shows the curved lane mapping result, and 11(b) displays the merged and forked lane mapping results. Point clouds are rendered with point intensity values. The dashed and continuous lane polylines are rendered in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

methods, consisting of ResNet-34 (He et al., 2016) for BEV image encoding and a basic vision transformer (Dosovitskiy et al., 2020) for capturing global features. The lane marking semantic segmentation employs the Feature Pyramid Network (FPN) (Lin et al., 2017) along with ResNet-34 (He et al., 2016) for BEV image encoding. Subsequently, skeletonization is applied to the results obtained from the semantic segmentation in quantitative evaluation. From Table 4, we can see that our method outperforms the other two methods in terms of all three evaluation metrics.

Compared to KLane and the traditional approach, the proposed method can produce lanes with richer attributes (both instance and semantic information). KLane has the capability to predict lane instances that lack marking semantics, whereas the classic method can only predict unstructured semantic markings without instance information. Our method detects vectorized lane markings with the awareness of both instances and semantics in an end-to-end manner.

Compared to KLane (Paek et al., 2022), our method does not impose any restrictions on the number of extracted lanes, as it is only limited by the number of column proposals, which far exceeds the actual lane count. In contrast, KLane establishes a correlation between lane numbers and decoding headers, whereby an increase in detected lane numbers is directly proportional to an increase in the number of decoding headers. Moreover, our BEV image generation procedure ensures that most lanes appear vertical in the BEV images. This makes our lane mapping pipeline independent of lane distribution such as the symmetrical lane distribution made by Li et al. (2022a), as exemplified in Fig. 11.

Fig. 12 presents the qualitative comparison between the proposed method and the two aforementioned methods. In the classic FPN-based semantic segmentation (Lin et al., 2017), skeletonization is applied to the segmented lane markings, and the extracted markings do not include instance information. KLane (Paek et al., 2022), on the contrary, extracts lane markings that are aware of lane instances but lack semantic information. In contrast, the proposed method extracts lane markings that not only incorporate both instance and semantic information but also show improved continuity and completeness.

4.6. Ablation study

The effectiveness of the hierarchical lane attention mechanism has also been validated by an ablation study, where we compare the performance of the proposed network with and without specific augmentations, including the ViT (Dosovitskiy et al., 2020) feature encoder and spatial attention, on both datasets B and C. Specifically, we replace the ViT feature encoder with the alternative feed-forward network with the same number of layers, and the auxiliary semantic segmentation header in column proposals is dropped when spatial attention is not adopted in the network. Table 5 presents the ablation results. From these ablation comparisons, we can see that the hierarchical lane attention mechanism enables to capture lane marking features and benefit the overall lane mapping performance. Besides, the inclusion of the spatial attention module leads to improved performance of the network on both datasets, particularly in terms of precision.

5. Conclusion

This paper proposes an end-to-end method for large-scale lane mapping from MLS point clouds. The main contributions of the study can be summarized into three points. Firstly, we introduce hierarchical lane attention, from the vision transformer at a global scale to spatial attention inside the local column proposal. This mechanism strengthens the robustness of the network against blurry lanes and incomplete data in the input point clouds, resulting in more accurate and complete lane mapping results. Secondly, the proposed lane mapping network effectively handles both geometrical lane polylines and their semantics, without requiring complex post-processing. Lastly, we provide a large-scale lane mapping dataset, comprising ground point clouds, BEV images, and corresponding annotations, which can stimulate research in automatic lane mapping. Comprehensive experimental results demonstrate the effectiveness of the proposed method for large-scale

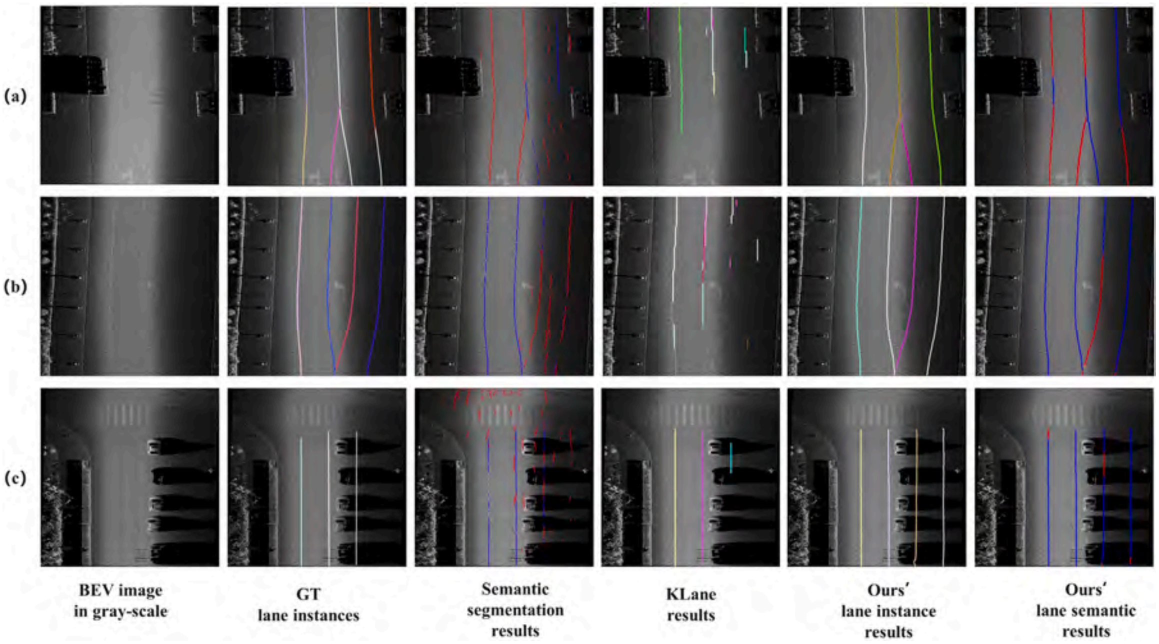

(a)
(b)
(c)
BEV image in gray-scale
GT lane instances
Semantic segmentation results
KLane results
Ours' lane instance results
Ours' lane semantic results

Fig. 12. Qualitative comparison results between the proposed method and classic FPN-based semantic segmentation (Lin et al., 2017) and KLane (Paek et al., 2022). For clarity, the lane markings are overlaid on grayscale images converted from the original BEV images. Lane instances are randomly colored. In terms of lane semantics, dashed lane markings are depicted in red, while continuous lane markings are in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Ablation study on datasets B and C. The ✓ and *x* signs indicate *with* and *without* a component, respectively. ViT refers to the vision transformer feature encoder (Dosovitskiy et al., 2020), and SA denotes spatial attention.

Augmentation			Dataset B									Dataset C								
ViT	SA		Buffer (5 pixel)			Buffer (10 pixel)			Buffer (20 pixel)			Buffer (5 pixel)			Buffer (10 pixel)			Buffer (20 pixel)		
			Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
x	x	Geometry	0.806	0.849	0.827	0.856	0.903	0.879	0.867	0.925	0.895	0.789	0.731	0.759	0.874	0.811	0.841	0.894	0.858	0.876
		Semantics	0.753	0.786	0.769	0.808	0.846	0.827	0.821	0.872	0.846	0.714	0.663	0.687	0.818	0.761	0.788	0.843	0.809	0.826
x	✓	Geometry	0.776	**0.867**	0.819	0.825	**0.921**	0.871	0.839	**0.945**	0.889	0.759	0.762	0.761	0.834	**0.838**	0.836	0.856	**0.879**	0.868
		Semantics	0.741	0.819	0.778	0.792	**0.876**	0.832	0.808	**0.901**	0.852	0.697	0.702	0.699	0.788	**0.791**	0.790	0.815	**0.832**	0.823
✓	x	Geometry	0.796	0.864	0.829	0.842	0.913	0.876	0.854	0.936	0.893	0.786	0.748	0.766	0.857	0.818	0.837	0.883	0.865	0.874
		Semantics	0.758	0.814	0.785	0.807	0.866	0.836	0.822	0.889	0.854	0.730	0.694	0.711	0.815	0.777	0.796	0.846	0.821	0.833
✓	✓	Geometry	**0.832**	0.852	**0.842**	**0.876**	0.898	**0.887**	**0.883**	0.917	**0.900**	**0.815**	**0.768**	**0.791**	**0.885**	0.835	**0.859**	**0.903**	0.873	**0.888**
		Semantics	**0.787**	**0.801**	**0.794**	**0.833**	0.848	**0.841**	**0.842**	0.870	**0.856**	**0.748**	**0.706**	**0.726**	**0.831**	0.785	**0.807**	**0.854**	0.821	**0.837**

lane mapping, even in challenging urban scenarios where moving objects cause occlusions in the source data. The generated lane maps can be directly integrated into city maps, enabling road infrastructure management and supporting intelligent transportation system. In the context of this study, our primary emphasis lies in the precise delineation of geometrical lane mapping. Nonetheless, it is important to acknowledge that the comprehensive understanding of large-scale lane topology holds significant implications for various aspects, including route planning and navigation. Future work will exploit lane topology, from intersections to the large-scale area, for detailed lane mapping.

CRediT authorship contribution statement

Xiaoxin Mi: Writing – original draft, Formal analysis. **Zhen Dong:** Writing – review & editing, Supervision. **Zhipeng Cao:** Resources. **Bisheng Yang:** Supervision. **Zhen Cao:** Validation. **Chao Zheng:** Funding acquisition. **Jantien Stoter:** Conceptualization. **Liangliang Nan:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by CSC (China Scholarship Council), the National Natural Science Foundation of China Projects under Grants No. 42130105, and Key Laboratory of Spatial–temporal Big Data Analysis and Application of Natural Resources in Megacities, MNR (No. KFKT-2022-01). The authors thank the Shanghai Surveying and Mapping Institute, Shanghai, China, for providing experimental data.

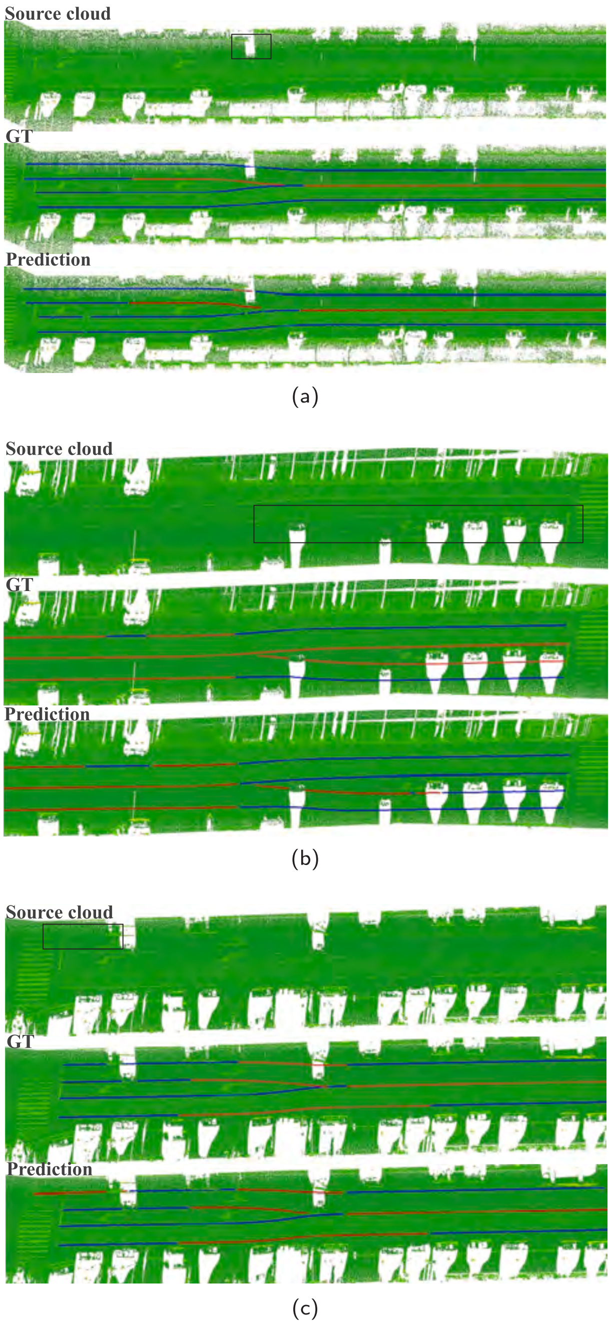


Fig. 13. A few failure cases. Point clouds are rendered with point intensity values. The dashed and continuous lane polylines are rendered in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

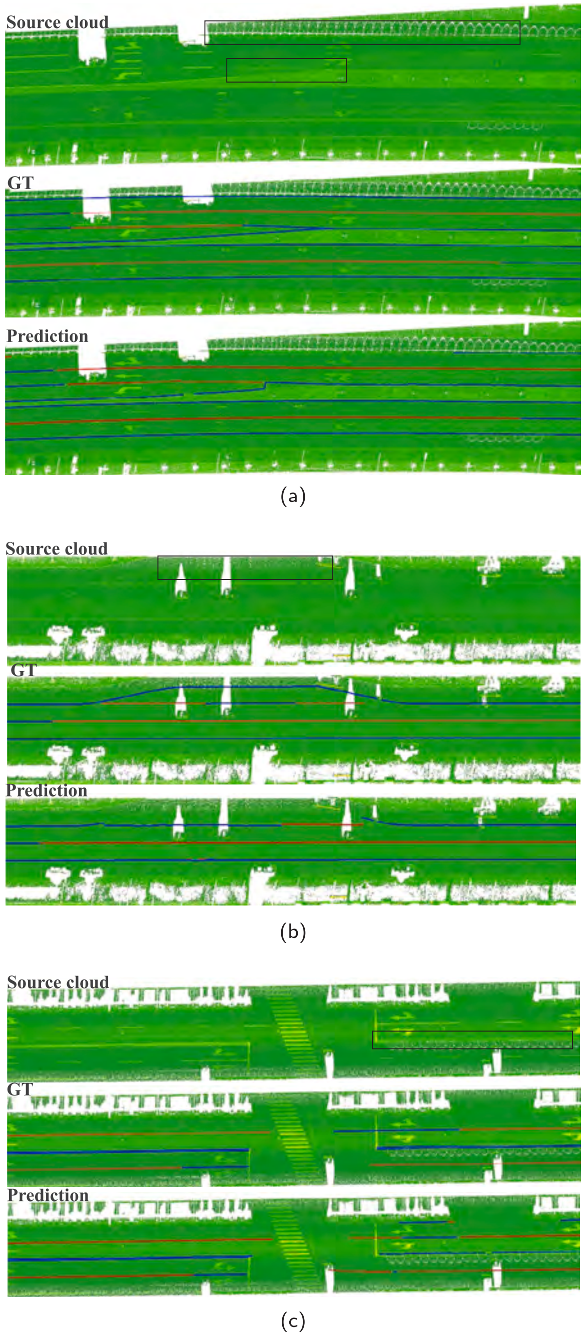


Fig. 14. Illustrations of failure cases in lane mapping. Point clouds are rendered with descending intensity, from yellow to green; dashed and continuous lane polylines are rendered in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

Bar Hillel, A., Lerner, R., Levi, D., Raz, G., 2014. Recent progress in road and lane detection: a survey. Mach. Vis. Appl. 25 (3), 727–745.

Borkar, A., Hayes, M., Smith, M.T., 2009. Robust lane detection and tracking with ransac and kalman filter. In: 2009 16th IEEE International Conference on Image Processing. ICIP, IEEE, pp. 3261–3264.

Cao, Z., Zhang, W., Wen, X., Dong, Z., Liu, Y.-S., Xiao, X., Yang, B., 2023. KT-Net: knowledge transfer for unpaired 3D shape completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37, pp. 286–294.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Choi, S., Kim, D., Park, S., Paik, J., 2022. Point cloud-based lane detection for optimal local path planning. In: 2022 International Conference on Electronics, Information, and Communication. ICEIC, IEEE, pp. 1–4.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Efrat, N., Bluvstein, M., Oron, S., Levi, D., Garnett, N., Shlomo, B.E., 2020. 3D-lanenet+: Anchor free lane detection using a semi-local representation. arXiv preprint arXiv:2011.01535.

Feng, Z., Guo, S., Tan, X., Xu, K., Wang, M., Ma, L., 2022. Rethinking efficient lane detection via curve modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17062–17070.

Garnett, N., Cohen, R., Pe'er, T., Lahav, R., Levi, D., 2019. 3D-lanenet: end-to-end 3d multiple lane detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2921–2930.

He, S., Balakrishnan, H., 2022. Lane-level street map extraction from aerial imagery. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2080–2089.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hervieu, A., Soheilian, B., Brédif, M., 2015. Road marking extraction using a model&data-driven rj-mcmc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 2 (3), 47.

Homayounfar, N., Ma, W.-C., Lakshmikanth, S.K., Urtasun, R., 2018. Hierarchical recurrent attention networks for structured online maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3417–3426.

Homayounfar, N., Ma, W.-C., Liang, J., Wu, X., Fan, J., Urtasun, R., 2019. Dagmapper: Learning to map by discovering lane topology. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2911–2920.

Jin, D., Park, W., Jeong, S.-G., Kwon, H., Kim, C.-S., 2022. Eigenlanes: Data-driven lane descriptors for structurally diverse lanes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17163–17171.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, H.T., Todd, Z., Bielski, N., Carroll, F., 2022b. 3D lidar point-cloud projection operator and transfer machine learning for effective road surface features detection and segmentation. Vis. Comput. 38 (5), 1759–1774.

Li, Q., Wang, Y., Wang, Y., Zhao, H., 2021. Hdmapnet: An online hd map construction and evaluation framework. arXiv e-prints, arXiv–2107.

Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L., 2022a. Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Lin, C., Guo, Y., Li, W., Liu, H., Wu, D., 2021. An automatic lane marking detection method with low-density roadside LiDAR data. IEEE Sens. J. 21 (8), 10029–10038.

Liu, L., Chen, X., Zhu, S., Tan, P., 2021a. Condlanenet: a top-to-down lane detection framework based on conditional convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3773–3782.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Mi, X., Yang, B., Dong, Z., Liu, C., Zong, Z., Yuan, Z., 2021b. A two-stage approach for road marking extraction and modeling using MLS point clouds. ISPRS J. Photogramm. Remote Sens. 180, 255–268. http://dx.doi.org/10.1016/j.isprsjprs.2021.07.012, URL: https://www.sciencedirect.com/science/article/pii/S0924271621001970.

Mi, L., Zhao, H., Nash, C., Jin, X., Gao, J., Sun, C., Schmid, C., Shavit, N., Chai, Y., Anguelov, D., 2021a. HDMapGen: A hierarchical graph generative model of high definition maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4227–4236.

Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L., 2018. Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 286–291.

Paek, D.-H., Kong, S.-H., Wijaya, K.T., 2022. K-lane: Lidar lane dataset and benchmark for urban roads and highways. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4450–4459.

Qin, Z., Wang, H., Li, X., 2020. Ultra fast structure-aware deep lane detection. In: European Conference on Computer Vision. Springer, pp. 276–291.

Qin, Z., Zhang, P., Li, X., 2022. Ultra fast deep lane detection with hybrid anchor driven ordinal classification. IEEE Trans. Pattern Anal. Mach. Intell..

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252.

Soilán, M., Nóvoa, A., Sánchez-Rodríguez, A., Justo, A., Riveiro, B., 2021. Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data. Autom. Constr. 126, 103684.

Soilán, M., Tardy, H., González-Aguilera, D., 2022. Deep learning-based road segmentation of 3D point clouds for assisting road alignment parameterization. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 43, 283–290.

Strudel, R., Garcia, R., Laptev, I., Schmid, C., 2021. Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7262–7272.

Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T., 2021. Keep your eyes on the lane: Real-time attention-guided lane detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 294–302.

Tang, J., Li, S., Liu, P., 2021. A review of lane detection methods based on deep learning. Pattern Recognit. 111, 107623.

Tardy, H., Soilán, M., Martín-Jiménez, J.A., González-Aguilera, D., 2023. Automatic road inventory using a low-cost mobile mapping system and based on a semantic segmentation deep learning model. Remote Sens. 15 (5), http://dx.doi.org/10.3390/rs15051351, URL: https://www.mdpi.com/2072-4292/15/5/1351.

Tian, W., Ren, X., Yu, X., Wu, M., Zhao, W., Li, Q., 2022. Vision-based mapping of lane semantics and topology for intelligent vehicles. Int. J. Appl. Earth Obs. Geoinf. 111, 102851. http://dx.doi.org/10.1016/j.jag.2022.102851, URL: https://www.sciencedirect.com/science/article/pii/S156984322200053X.

Tian, Z., Shen, C., Chen, H., He, T., 2019. Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9627–9636.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., Habib, A., 2019. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 147, 178–192. http://dx.doi.org/10.1016/j.isprsjprs.2018.10.007, URL: https://www.sciencedirect.com/science/article/pii/S0924271618302855.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. Adv. Neural Inf. Process. Syst. 34, 12077–12090.

Yang, B., Fang, L., Li, Q., Li, J., 2012. Automated extraction of road markings from mobile LiDAR point clouds. Photogramm. Eng. Remote Sens. 78 (4), 331–338.

Yu, Y., Li, J., Guan, H., Jia, F., Wang, C., 2014. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (2), 709–726.

Zang, A., Xu, R., Li, Z., Doria, D., 2017. Lane boundary extraction from satellite imagery. In: Proceedings of the 1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles. pp. 1–8.

Zheng, F., Luo, S., Song, K., Yan, C.-W., Wang, M.-C., 2018. Improved lane line detection algorithm based on hough transform. Pattern Recognit. Image Anal. 28, 254–260.

Zhou, Y., Takeda, Y., Tomizuka, M., Zhan, W., 2021. Automatic construction of lane-level hd maps for urban scenes. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 6649–6656.