This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning to Detect 3D Lanes by Shape Matching and Embedding

Ruixin Liu, Zhihao Guan, Zejian Yuan
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
{sweetylrx, duduguan}@stu.xjtu.edu.cn   yuan.ze.jian@xjtu.edu.cn

Ao Liu, Tong Zhou, Tang Kun, Erlong Li, Chao Zheng, Shuqi Mei
T Lab, Tencent Map, Tencent, China
{allliu, kyriezhou, kunntang, erlongli, chrisczheng, shawnmei}@tencent.com

Abstract

3D lane detection based on LiDAR point clouds is a challenging task that requires precise locations, accurate topologies, and distinguishable instances. In this paper, we propose a dual-level shape attention network (DSANet) with two branches for high-precision 3D lane predictions. Specifically, one branch predicts the refined lane segment shapes and the shape embeddings that encode the approximate lane instance shapes, the other branch detects the coarse-grained structures of the lane instances. In the training stage, two-level shape matching loss functions are introduced to jointly optimize the shape parameters of the two-branch outputs, which are simple yet effective for precision enhancement. Furthermore, a shape-guided segments aggregator is proposed to help local lane segments aggregate into complete lane instances, according to the differences of instance shapes predicted at different levels. Experiments conducted on our BEV-3DLanes dataset demonstrate that our method outperforms previous methods.

1. Introduction

3D lane detection based on LiDAR point clouds is an essential visual-perception task for autonomous driving, which provides centimeter-level locations, exact geometric shapes, and instance-level information of ego and neighboring lanes. Great attention has been attracted due to its numerous real-world applications such as lane departure warning [19], lane keeping assistance, vehicle navigation, and high definition (HD) map construction [9].

Recently, high-precision 3D lane detection remains challenging. Most previous studies are conducted on RGB images [22, 10, 14, 4, 12], typically based on the front view. However, the camera's inherent optical sensitivity and distortion limit the precision of detection and the alignment from images to 3D space. Besides, lanes exist in various slender shapes (straight lines, polylines, curves, etc.) and complex topologies (forks, merges, etc.), making occlusions from the front view common circumstances. In contrast to images, point clouds can fully preserve the accurate raw geometric information of 3D scenarios, therefore, the sparse and irregular point clouds are encoded to the structured bird's-eye view feature maps for 3D lane detection, following [2, 1, 8, 9].

Existing studies have achieved significant developments through deep learning techniques. Segmentation-based methods [22, 30, 17, 21] infer pixel-wise segmentation maps, requiring dense annotations and cumbersome post-processings. Anchor-based [13, 5, 26, 24] methods predict lane shapes with reference to predefined anchors, which makes the predictions away from the anchors inaccurate. Parametric-based [15, 27, 4] methods design holistic lane shape models, with swift inference and no post-processing, but with less superiority in precision. As an alternative, grid-based methods [23, 18, 14, 10] eliminate redundant predictions by flexible sparse point regression. Such methods cluster lane points by measuring the similarity of embedding features [10] or the features with the highest spatial correlation [24]. However, the random initial clustering centers have a significant impact on the clustering results.

In this work, we design a dual-level shape attention network (DSANet) with two branches, which focuses on both the local structures of the lane and the global lane shapes. One branch detects the lane *segment shapes*, and predicts the corresponding *shape embeddings* that encode the approximate shapes of the lanes. The other branch focuses on the global information of the lanes and predicts the lane *instance shapes*. Besides, shape matching and embedding loss functions are proposed, which can jointly optimize shape parameters and effectively improve training speed and lane fitting accuracy. More significantly, a local lane segments clustering method guided by global instance shapes (i.e., *shape-guided segments aggregator*) is proposed to utilize the outputs of our DSANet to effectively aggregate the segment shapes and determine the number of lane instances.

We validate the effectiveness of our method by conducting comparative experiments and ablation studies on the self-collected 3D point cloud dataset BEV-3DLanes, under stricter centimeter-level distance thresholds. Experiments on our dataset demonstrate that our method outperforms previous state-of-the-art methods with excellent accuracy, especially under higher precision criteria. The main contributions of our work can be summarized as follows:

- We develop a novel dual-level shape attention network (DSANet) with two branches that fuses the contextual information at the local and global levels to detect high-precision 3D lanes.

- Simple and effective dual-level representations of lane shapes and the corresponding shape matching constraints are proposed to separately predict the fine-grained segment shapes and the coarse-grained instance shapes.

- A shape-guided segments aggregator is designed to cluster the flexible segments into instances, with the instance shapes serving as the explicit clustering centers.

2. Related Work

This section discusses existing LiDAR-based and image-based lane detection methods. Traditional methods mostly depend on hand-crafted features (based on color [3], intensity value [7], edge [11], etc.) and strong prior assumptions [16], lacking robustness to variable scenarios. Recently, deep-learning-based methods have shown superior performances in terms of both accuracy and speed, which can be roughly classified into the following paradigms.

Segmentation-based methods. These methods for images [2, 22, 20, 30] and point clouds [17, 21] consider lane detection as pixel-wise semantic segmentation tasks, which assigns each pixel a label of lane or non-lane. LoDNN [2] leverages FCNs to enlarge the receptive field and predicts a road confidence map for the top-view images. SCNN [22] and LLDN-GFC [21] consider lane detection as multi-class segmentation problems, predicting a probability map for each lane category, and using heuristic techniques for clustering and lane fitting. Segmentation-based methods require dense annotations, redundant predictions and cumbersome post-processings, with considerable time and computational costs. In comparison, our method simplifies the architecture to a lightweight yet efficient form with grid-level predictions and accurate shape matching.

Grid-based methods. Such methods follow a grid-wise manner in analogy to semantic segmentation, combined with fine-grained localization regression inside each grid. The predictions are flexible due to the weak inter-grid constraints. UFAST [23] formulates lane detection to a row-based selecting method with global features, achieving a swift speed. CondLaneNet [14] proposes a top-to-down framework that utilizes proposal head and conditional convolution to acquire instance discrimination. Inspired by human pose estimation, PINet [10] utilizes stacked hourglass networks to predict grid-wise key points and embedding features, and clusters points into instances by measuring the similarities between embedding features. HRAN [8] iteratively predicts the initial regions and the vertex sequences of lane boundaries from a top-to-down BEV LiDAR image, exploiting a hierarchical recurrent network. Most grid-based methods cannot avoid post-processings. To cluster sparse points into individual instances, they measure the similarity of implicit embedding features without physical meanings or iteratively search the neighbors by features with the highest spatial correlation. As an improvement, the abundant local geometry information of our method is integrated to predict a precise lane segment shape for each grid, and a low-dimension feature that embeds the corresponding instance shape is designed for grouping.

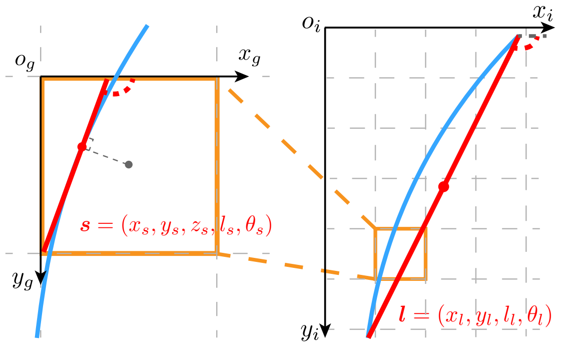
Other methods. Some methods like Line-CNN [13], 3D-LaneNet [5] and LaneATT [26] regress the offsets to optimize lane shape with pre-designed anchors. Other methods like PolyLaneNet [27] and LSTR [15] model the lane shape by a polynomial representation, and predict parameters for the reformulated polynomial regression problem. Besides, DAGMapper [9] formulates a directed acyclic graphical model (DAG) to recurrently infer the localization, topology and state of lane within the rotated region of interest. Some of these methods are post-processing free, but the strong shape priori assumptions prevent them from fitting complex scenarios well. Our DSANet flexibly regresses an accurate segment shape for each grid, which is applicable for complex topologies and lanes with high curvatures.

3. Methodology

This section defines the lane detection problem following parts-to-whole strategies: detecting small segments and grouping them into complete instances, where both local and global contexts are considered. Concretely, we propose a bottom-up method that concentrates on lane shapes at two levels, and we introduce shape matching and embedding loss functions for joint optimization. Besides, the shape-guided segments aggregator that clusters segments into instances is illustrated.

3.1. Dual-level Lane Shape Representations

The dual-level lane shape representations describe the local fine-grained segment shapes and the global coarse-grained instance shapes, respectively.


o_g
x_g
y_g
$\boldsymbol{s} = (x_s, y_s, z_s, l_s, \theta_s)$
o_i
x_i
y_i
$\boldsymbol{l} = (x_l, y_l, l_l, \theta_l)$

(a) (b)

Figure 1. Dual-level lane shape representations, marked in red. (a): Segment shape representation is the straight line segment that approximates the local lane inside the grid, defined in the grid coordinate system x_g-o_g-y_g. (b): Instance shape representation is the straight line segment that connects the start and the end of the complete lane, defined in the image coordinate system x_i-o_i-y_i.

Segment shape representation. The local lane segment shape is defined with a flexible grid-wise representation. Given a pseudo-BEV image $I \in R^{H \times W \times 4}$ projected from aggregated LiDAR point clouds, I is divided into grids with the size of $r \times r$, generating a total of $\frac{H}{r} \times \frac{W}{r}$ non-overlapping grids. For each grid, there may exist a small lane segment, represented by a vector $\boldsymbol{s} = (x_s, y_s, z_s, l_s, \theta_s)$, where x_s and y_s are the offsets to the origin of grid, l_s and θ_s are the length and radian, and z_s is the elevation in 3D space, as shown in Figure 1 (a), marked in red. Specifically, the lane segment is approximated as a straight line segment passing through (x_s, y_s), which is obtained by sampling the point closest to the grid center. l_s and θ_s are the properties of the red tangent line. Compared to the common grid-based methods that directly detect sparse points, the segment shape representation can supplement more local lane shape details.

Shape embedding representation. For each grid, 4-channel shape embeddings $\boldsymbol{e} = (x_e, y_e, l_e, \theta_e)$ associated with the segment shape are introduced to encode the shape appearances of distinguishable instances into a low-dimension feature space with physical meaning, which specifies the instance that the segment may belong to. (x_e, y_e) localizes the center point of the straight line segment that approximates the lane instance, and l_e and θ_e are the length and radian. Compared with the frequently used embedding features [10, 24], $\boldsymbol{e}$ is more rational due to the joint constraints on the physical instance shapes.

Instance shape representation. Since a coarse description is sufficient to distinguish the lane instances, the lane instance shape is approximated as a straight line segment that connects its start and end, as shown in Figure 1 (b), marked in red. A global lane instance can be represented as $\boldsymbol{l} = (x_l, y_l, l_l, \theta_l)$, where (x_l, y_l) are the center coordinates of the straight line segment w.r.t. the image coordinate system, and l_l and θ_l are its length and radian.

3.2. Dual-Level Shape Attention Architecture

The overall architecture of our proposed DSANet is illustrated in Figure 2, which contains a shared backbone composed of CNNs and self-attention (SA) [15] modules, and two branches for the two-grained lane shapes.

Backbone. Our backbone consists of CNNs and self-attention modules. Given I, ResNet18 [6] is used to extract low-resolution features $\mathcal{F}_c$ that encode high-dimension spatial information. Since lanes require global contextual information due to their slender shapes and long ranges, self-attention modules are stacked after CNNs to better capture global correlation. $\mathcal{F}_c$ is transmitted to self-attention modules to supplement the fused features with positional information, and the output sequence of the self-attention modules is reshaped as $\mathcal{F}_s$ for the two branches.

Segment shape branch. The segment shape branch regresses the grid-level segment shapes and shape embeddings by two decoupled heads using a refined feature map. On the one hand, the lane segment (LS) head predicts accurate intra-grid lane locations and segment shapes. More specifically, the LS head outputs a grid-level map of size $\frac{H}{r} \times \frac{W}{r} \times 6$, where the first channel is a set of segment existence confidences C_s to indicate whether there exists a lane segment in each grid, and the rest channels separately correspond to the 5 shape parameters of the segment $\boldsymbol{s}$, whose set is written as S. On the other hand, the shape embedding (SE) head predicts shape parameter estimations of the lane instances that the segments may belong to, denoted as E.

Instance shape branch. The instance shape branch predicts the global instance shapes through several CNNs and a feed-forward network (FFN) with a simple 2-layer perception. To predict a variable number ($< N$) of lane instances, the output of the instance shape branch is fixed as $(1+4)N$, where $1N$ indicates the instance existence confidences of N lane instances, written as C_l. And $4N$ separately corresponds to the shape parameters of the possible lane instances, denoted as L.

3.3. Training with Shape Matching and Embedding Constraints

DSANet is trained with shape matching and embedding loss functions, which separately supervise segment shapes, shape embeddings and instance shapes between predictions and ground truths. The total loss function is formulated as:

$$L_{total} = L_{seg\_sm} + L_{embed} + L_{ins\_sm}. \tag{1}$$

Segment shape matching loss. Segment shape matching loss L_{seg_sm} imposes constraints on existence confidences, fine-grained locations and geometry shapes of the

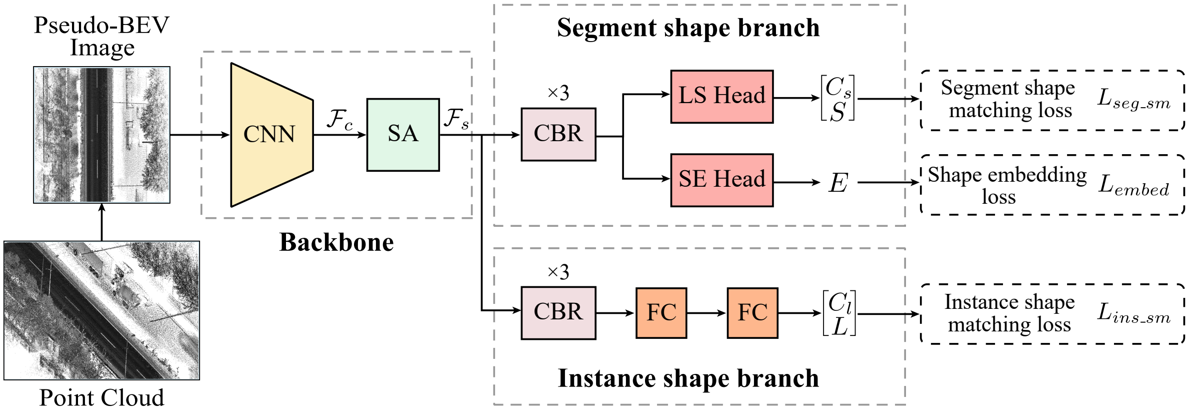

Pseudo-BEV Image
Point Cloud
CNN
$\mathcal{F}_c$
SA
$\mathcal{F}_s$
Backbone
Segment shape branch
×3
CBR
LS Head
$\begin{bmatrix} C_s \\ S \end{bmatrix}$
SE Head
E
Segment shape matching loss L_{seg_sm}
Shape embedding loss L_{embed}
×3
CBR
FC
FC
$\begin{bmatrix} C_l \\ L \end{bmatrix}$
Instance shape matching loss L_{ins_sm}
Instance shape branch

Figure 2. Network architecture. The raw point clouds are rasterized and projected into a pseudo-BEV image I, serving as the input to our DSANet. The shared backbone consisting of CNNs and self-attention (SA) modules is used for feature extraction. $\mathcal{F}_s$ is fed to both segment shape branch and instance shape branch, which are made of convolution, batch normalization and ReLU (CBR), as well as fully connected (FC) layers. The segment shape branch outputs the segment existence confidences (C_s) and high-precision segment shapes (S) by the LS head, and the SE head outputs grid-level shape embeddings (E). The instance shape branch outputs instance existence confidences (C_l) and instance shapes (L). L_{seg_sm}, L_{embed} and L_{ins_sm} respectively supervise the above outputs.

local segments as follows:

$$L_{seg\_sm} = \gamma_c L_{lconf} + \gamma_o L_{offset} + \gamma_{sm} L_{lsm}, \tag{2}$$

where L_{lconf}, L_{offset}, and L_{lsm} serve as segment existence confidence loss, offset loss and local shape matching loss, respectively. The weights γ_c, γ_o and γ_{sm} are used to balance the effects of loss terms. Segment existence confidence loss L_{lconf} separately constrains grids with and without segments, following [10]. Offset loss L_{offset} minimizes the differences between the ground truths and the predictions of the positive samples by measuring the relative elevation offsets, implemented by smooth-L1.

L_{lsm} constrains the segment shapes inside the same grids, following [29]. Instead of constraining the parameters independently, the arbitrary-oriented lane segment $\boldsymbol{s}$ is converted into a 2D Gaussian distribution $\mathcal{N}_s(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ for supervision. $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ are the mean and covariance of $\mathcal{N}_s$, written as:

$$\boldsymbol{\mu}_s = (x_s, y_s)^T, \tag{3}$$

$$\begin{aligned} \boldsymbol{\Sigma}_s^{\frac{1}{2}} =& \boldsymbol{R\Lambda R}^T = \begin{pmatrix} \cos\theta_s & -\sin\theta_s \\ \sin\theta_s & \cos\theta_s \end{pmatrix} \cdot \\ & \begin{pmatrix} w_s & 0 \\ 0 & \frac{l_s}{2} \end{pmatrix} \cdot \begin{pmatrix} \cos\theta_s & \sin\theta_s \\ -\sin\theta_s & \cos\theta_s \end{pmatrix}, \end{aligned} \tag{4}$$

where R represents the rotation matrix, Λ is the diagonal matrix, x_s, y_s, l_s and θ_s are properties of $\boldsymbol{s}$, and $w_s = \frac{l_s}{6}$ is regarded as the width of the lane segment.

Given the positive samples of lane segments $S^+ \subseteq S$, the prediction $\mathcal{N}_s$ and the ground truth $\bar{\mathcal{N}}_s$ are forced to close to each other by L_{lsm}:

$$L_{lsm} = \frac{1}{|S^+|} \sum_{\boldsymbol{s} \in S^+} \xi\left(\bar{\mathcal{N}}_s, \mathcal{N}_s\right), \tag{5}$$

where $|\cdot|$ denotes the cardinality of the set, and the ground truth $\bar{\mathcal{N}}_s$ is constructed following Eq. 3 and Eq. 4. It has been mathematically proven in [29] that a larger aspect ratio will result in more attention on the angle, making L_{lsm} better meet the prediction demands of lane direction.

$\xi\left(\bar{\mathcal{N}}_s, \mathcal{N}_s\right)$ is implemented by the symmetric Kullback-Leibler divergence (KLD), which constructs a chain coupling relationship composed of all parameters, making the joint optimization a self-modulated mechanism. The symmetric KLD between the two 2D Gaussian distributions is denoted as:

$$\begin{aligned} \xi(\bar{\mathcal{N}}_s, \mathcal{N}_s) = \frac{1}{2} \Big(& KLD\left(\bar{\mathcal{N}}_s, \mathcal{N}_s\right) + \\ & KLD\left(\mathcal{N}_s, \bar{\mathcal{N}}_s\right) \Big). \end{aligned} \tag{6}$$

Shape embedding loss. Shape embedding loss L_{embed} constrains the lane instance shapes detected from the local feature map, written as:

$$L_{embed} = \frac{1}{|E^+|} \sum_{\boldsymbol{e} \in E^+} \xi\left(\bar{\mathcal{N}}_e, \mathcal{N}_e\right), \tag{7}$$

where $E^+ \subseteq E$ is the set of positive samples, and e is the shape embeddings. Similar to the definition introduced before, the 2D Gaussian distributions of the prediction $\mathcal{N}_e$ and the ground truth $\bar{\mathcal{N}}_e$ are constructed using the 4 properties of the shape embeddings.

Instance shape matching loss. Instance shape matching loss L_{ins_sm} is for the instance shape branch, which supervises coarse-grained shape information of instances, written as:

$$L_{ins\_sm} = \gamma_c L_{gconf} + \gamma_o L_{gsm}, \tag{8}$$

where instance confidence loss L_{gconf} supervises the existence confidences of the instances, and global shape matching loss L_{gsm} constrains the shape parameters of the instances. Given the positive samples of lane instances $L^+ \subseteq L$, L_{gsm} is written as:

$$L_{gsm} = \frac{1}{|L^+|} \sum_{\hat{\Phi}(\boldsymbol{l}) \in L^+} \xi \left(\bar{\mathcal{N}_{\boldsymbol{l}}}, \mathcal{N}_{\hat{\Phi}(\boldsymbol{l})} \right), \tag{9}$$

where $\boldsymbol{l}$ is the ground-truth lane instance, and $\hat{\Phi}(\boldsymbol{l})$ is the predicted lane instance associated with $\boldsymbol{l}$ by solving a bipartite matching problem. $\bar{\mathcal{N}_{\boldsymbol{l}}}$ and $\mathcal{N}_{\hat{\Phi}(\boldsymbol{l})}$ are their respective 2D Gaussian distributions.

Bipartite matching. The optimal assignment $\hat{\Phi}$ for the bipartite matching problem can be obtained by Hungarian algorithm [25, 15]. To construct the problem with N predictions and M ground truths ($M \leq N$), the ground truths are extended to N dimensions, with the last $N - M$ dimensions padded with zeros. Given the ground-truth lane instance set $\bar{L}$, the problem can be equivalently regarded as a cost minimization problem by searching a one-to-one mapping function $\Phi : \bar{L} \rightarrow L$:

$$\hat{\Phi} = \arg\min_{\Phi} \sum_{\boldsymbol{l} \in \bar{L}} \mathcal{C}\left(\boldsymbol{l}, \Phi(\boldsymbol{l})\right), \tag{10}$$

where $\mathcal{C}$ is the cost function that measures the spatial distance between the ground truth and the prediction.

3.4. Shape-guided Segments Aggregator

The outputs of the network, if the corresponding existence confidences are greater than the preset thresholds, are regarded as the positive predictions of segment shapes, shape embeddings E^+, and instance shapes L^+. E^+ and L^+ are fed to our proposed shape-guided segments aggregator, with L^+ serving as explicit clustering centers. Besides, each segment decides the instance it belongs to by independently voting for the instance shape L_i^+ closest to its shape embeddings E_j^+, where i and j are the respective indexes.

The correspondence between the indexes of E^+ and L^+ is defined as ϵ, and the similarity D between the j-th shape embeddings E_j^+ and the i-th instance shape L_i^+ is defined as $D\left(E_j^+, L_i^+\right)$, implemented by L1 norm. By minimizing D, the j-th segment is associated with the $\hat{\epsilon}(j)$-th instance, written as:

$$\hat{\epsilon}(j) = \arg\min_{i} D\left(E_j^+, L_i^+\right), \tag{11}$$

which is performed on all the elements of E^+ to obtain the final aggregation results $\hat{\epsilon}$. In contrast to the frequently used post-clustering methods, our proposed shape-guided segments aggregator ensures a relatively correct number of lanes by exploiting the detection results of the instance shape branch, and efficiently prevents the incorrect shift of cluster centers caused by randomly initialization.

4. Experiments

Dataset. Experiments are conducted on a self-collected BEV-3DLanes dataset due to the unavailability of existing large-scale public datasets. BEV-3DLanes is a large-scale real-world 3D lane dataset, which contains 200 km of roads in multiple cities with various traffic, lighting, and weather conditions. The 3D scenes are collected by LiDAR sensors and aggregated into high-density point clouds. For a single data frame, the number of points is around 400k, covering a length of around 4 m. The ground-truth labels are annotated in a local coordinate system in 3D space, including the localization and instance information. The adjacent 7 frames are spliced to obtain sufficient information about the slender lane structure, and the center of the spliced data is defined as the track information of 4th frame. Points beyond the fixed-size 3D window around the center are filtered out, where $x \in \{-12.5, 12.5\}$, $y \in \{-12.5, 12.5\}$, and $z \in \{-2, 1\}$, in meters. The point clouds within the window are rasterized and then projected onto a bird's-eye view (BEV) to generate a 4-channel (mean intensity, density, elevation difference, and minimum elevation) pseudo-BEV image dataset. There are in total 35277 data frames with a resolution of 800×800, split into 29874 for the training set and 5403 for the testing set.

Evaluation Metrics. We follow [28] to redefine the metrics precision (%), recall (%), and F1 score (%) in more rigorous manners, applicable for the evaluations of lane detection tasks based on point clouds. To be concrete, the predicted and the ground-truth lane points are interpolated to denser ones with the same number of outputs. The number of predicted points that fall within a threshold of the ground-truth points is leveraged to compute the evaluation metrics. Our experiments focus on the thresholds of 5 cm, 10 cm and 20 cm at physical distances, corresponding to images with 1.6 pixels, 3.2 pixels and 6.4 pixels.

Implementation details. All the images are centered crop to size 775×775 and resized to 512×512 as the input to the network. The height and width of the grid cell are set as 32 to balance the accuracy and costs. Besides, the inputs are augmented by scaling, rotating, and flipping. To avoid overfitting, the output channel of each block of ResNet18 is reduced to "32, 64, 128, 256". Then 6 self-attention modules are stacked after CNNs to obtain the shared feature map $\mathcal{F}_s$. In the training stage, loss coefficients γ_c, γ_o, and γ_{sm} are set to 2.0, 5.0, and 0.6. The fixed number of lane in-

Methods	Results (5 cm)			Results (10 cm)			Results (20 cm)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
CondLaneNet [14]	68.72	70.25	69.48	84.67	**86.35**	85.50	89.64	**92.43**	91.01
PINet [10]	61.08	59.07	60.06	85.14	83.41	84.27	91.29	89.83	90.55
PolyLaneNet [27]	50.71	49.93	50.32	72.92	72.43	72.67	86.30	86.07	86.18
LSTR [15]	57.42	57.24	57.33	79.70	80.07	79.88	90.18	90.82	90.50
Lane-ATT [26]	62.51	63.68	63.09	79.81	81.98	80.88	88.12	91.08	89.58
DSANet (Ours)	**74.24**	**71.40**	**72.79**	**87.41**	85.14	**86.26**	**92.10**	91.09	**91.59**

Table 1. Comparisons of precision (%), recall (%) and F1 score (%) on BEV-3DLanes testing set.

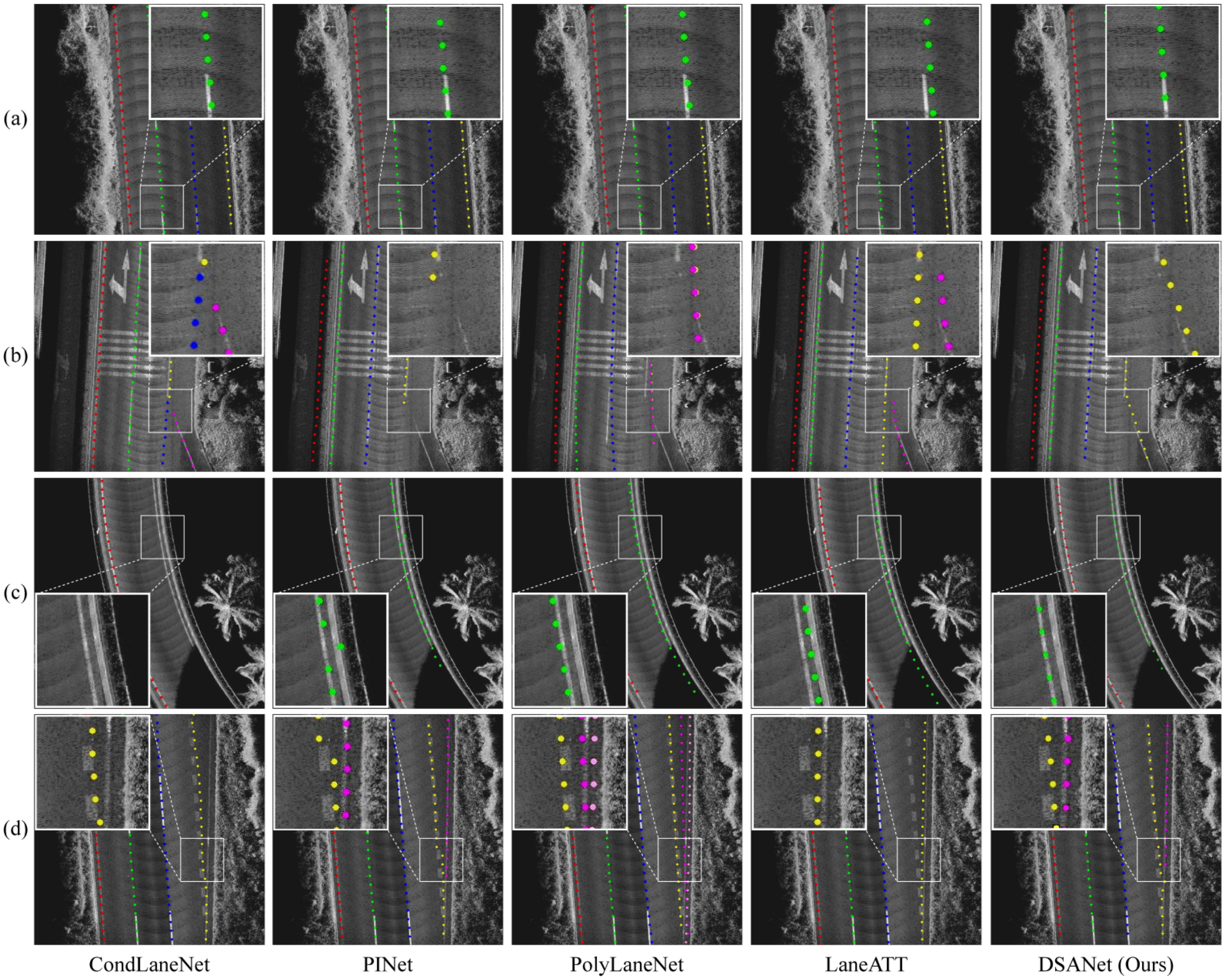


Figure 3. Visualizations of comparisons on BEV-3DLanes. (a): general cases. (b): polylines. (c): curves. (d): forks.

stances N is set as 10. For optimization, Adam algorithm is adopted with the initial learning rate set to be 0.0001, and a decay factor of 0.5 per 30 epochs starting from the 80th epoch. Batch size and training epochs are set to 64 and 150, respectively.

4.1. Comparisons with State-of-the-Art methods

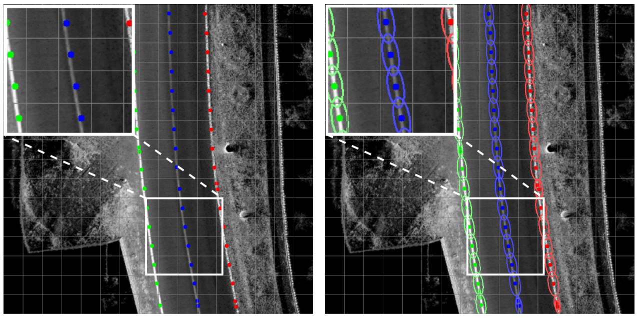
Baselines. Since few baselines with open-sourced codes and datasets are available for the 3D lane detection task based on BEV, we transform several methods designed for the analogous front-view lane detection tasks into workable forms, with the elevations directly set as ground truths. CondLaneNet [14], PINet [10], PolyLaneNet [27], LSTR [15] and Lane-ATT [26] are treated as our competitors, where LSTR requires polynomial curve modeling adaption from front view to BEV, and LaneATT requires setting for pre-designed anchors in experiments. During the evaluation, DSANet outputs 3D lanes instead of the 2D predictions of other methods. That is to say, our method is estimated under more rigorous criteria. Besides, DSANet can be applied in any orientation and with an arbitrary number of lanes, meeting the demands of various task scenarios.

Table 1 shows the comparative performances on BEV-3DLanes testing set. Our DSANet achieves the state-of-the-art results on F1 scores by a clear margin, especially in circumstances with tighter thresholds. To be specific, DSANet beats CondLaneNet with 3.3%, 0.8% and 0.6% F1 score boosts under 5 cm, 10 cm and 20 cm. Besides, DSANet achieves 12.7% and 9.7% higher F1 scores than PINet and LaneATT under 5 cm, and achieves 2.0% and 5.4% higher F1 scores under 10 cm. Furthermore, we observe that our DSANet attains a significant improvement over PolyLaneNet and LSTR for high precision demands. The improvement might be attributed to the segment shape matching constraints, which ensure the predictions simultaneously satisfy the inter-grid flexibility and the intra-grid precision. Besides, the shape-guided segments aggregator may cause a precision boost, confirmed by ablation studies.

Figure 3 illustrates the visualizations of some detection results in different scenarios on 2D images, including (a) general cases, (b) polylines, (c) curves, as well as (d) forks. The rightmost column substantiates that our method performs complete lane results with the highest precision. For the general cases and lanes with curves, our DSANet regresses flexible and accurate offsets for each grid with constraints on segment shapes. By contrast, CondLaneNet and PINet are inadequately precise, caused by the point outputs lacking shape constraints. Meanwhile, PolyLaneNet and LaneATT have no advantages in predicting accurate localization due to the limitations of the strong lane shape assumptions. As for situations with polylines and forks, our DSANet overwhelms other methods thanks to the correct number predictions of the lane instance shapes, which are realized by adjusting the proportion of positive and negative samples. In comparison, false positives (PolyLaneNet (b), (d) and LaneATT (b)) and false negatives (LaneATT(d)) are caused by incorrect predictions of the numbers of lane instances.

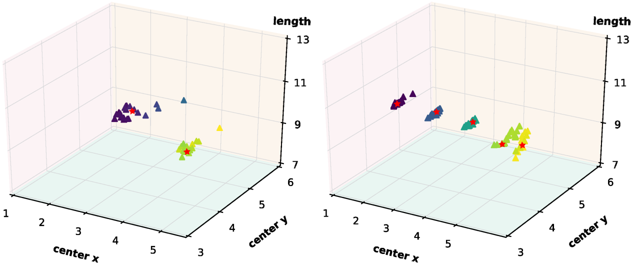
4.2. Ablation studies

Extensive ablation studies are conducted to analyze the contributions of each part, including segment shape matching (**Seg**) constraints, instance shape matching (**Ins**) and shape embedding (**SE**) constraints, and shape-guided segments aggregator (**Shape-guided**) for grouping. The results are shown in Table 2.



(a) PT representation (b) Seg representation

Figure 4. Visualizations of predictions based on PT and our Seg. The ellipses within grids are the visualizations of the 2D Gaussian distributions $\mathcal{N}_s$ as introduced in Section 3.3. It is worth noting that the ellipses are practically indistinguishable, and the colors in the figure are for better visual discrimination.



(a) Curve (b) Fork

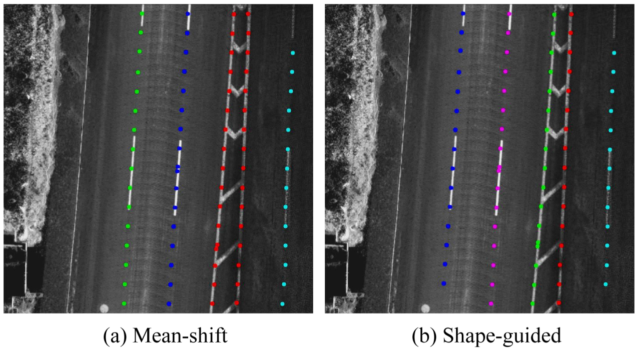
Figure 5. Visualizations of our proposed shape embeddings. The triangles and the red stars are $\boldsymbol{e}$ and $\boldsymbol{l}$, respectively. The colors and the sizes of the triangles are decided by x_e and θ_e, respectively.

Efficacy of segment shape matching constraints. To substantiate the contribution of Seg, the Seg constraints on lane segments $\boldsymbol{s}$ are replaced by smooth-L1 constraints on lane points (PT). The ablation results are listed in Table 2. It can be clearly observed that Seg improves the performances by 7.43% and 0.80% on F1 score over PT under 5 cm and 10 cm. This owes to the L_{lsm} coupling the parameters of the segments for supervision, by which more geometric information within the local grid can be provided, consequently yielding higher localization precision and more complete lane predictions. The visualizations of predictions based on PT and Seg are shown in Figure 4.

Importance of instance shape matching and shape embedding constraints. In this part, the importance of Ins and SE constraints is validated. Table 2 shows that removing Ins and SE results in a decrease in F1 score by 2.84% and 0.59% under 5 cm and 10 cm, respectively. The visualizations of the predicted shape embeddings are shown in Figure 5. The color of the triangle is given by its x_e, we can find that the shape embeddings of predicted segments be-

Grid		Instance		Aggregation strategies		Results (5 cm)			Results (10 cm)		
Seg	PT	Ins	SE	Mean-shift	Shape-guided	Precision	Recall	F1	Precision	Recall	F1
✓		✓	✓		✓	74.24	71.40	72.79	87.41	85.14	86.26
✓					✓	71.92	68.09	69.95(-2.84)	86.90	84.48	85.67(-0.59)
✓				✓		68.39	63.24	65.71(-7.08)	87.83	83.31	85.51(-0.75)
	✓	✓	✓		✓	67.39	63.45	65.36(-7.43)	86.87	84.10	85.46(-0.80)
	✓				✓	66.02	61.52	63.69(-9.10)	86.79	82.71	84.70(-1.56)
	✓			✓		60.82	60.74	60.78(-12.01)	84.13	84.90	84.51(-1.75)

Table 2. Ablation results. The constraints on the grid (Grid) include segment shape matching (Seg) and smooth-L1 loss on points (PT), the constraints on the instance (Instance) include instance shape matching (Ins) and shape embedding (SE), and the aggregation strategies include the widely used mean-shift and our shape-guided strategies.


(a) Mean-shift

(b) Shape-guided

Figure 6. Aggregation strategies based on mean-shift and our proposed shape-guided segments aggregator.

longing to the same instances are distributed close to each other, while those belonging to different instances are far apart.

Contribution of shape-guided segments aggregator. To verify the effectiveness of our proposed shape-guided segments aggregator, the widely used post-clustering strategy mean-shift is taken as a competitor. The results in Table 2 illustrates decreases of 4.24% and 0.16% under 5 cm and 10 cm distance thresholds. The visualized comparison results are shown in Figure 6. We notice that mean-shift aggregates the two close lanes into the same instance. The reason is that the quite similar shape embeddings make the mean value shift to an incorrect direction. In contrast to clustering without centers, DSANet leverages the instance shape branch to predict a set of lane instances, efficiently avoiding an inaccurate aggregating number of instances and providing explicit guidance for lane segments aggregation.

To summarize the ablation studies, we attribute precision enhancement to the shape-guided segments aggregator (7.08% under 5cm) jointly supported by instance shapes and shape embeddings, and the coupled shape matching constraints on segments (7.43% under 5cm). More intermediate and final results of comparative experiments and ablation studies can be visualized in the **Supp.**.

5. Conclusion

In this work, we propose a 3D lane detector with two-level lane shape predictions. The shape matching and embedding loss functions are introduced to improve the accuracy of unified shape representations, and the shape-guided segments aggregator effectively enhances the discrimination of lane instances. The whole framework is validated on the self-collected dataset, surpasses the previous methods in terms of high precision, and achieves clustering boosts. However, the correlation between the inter-frame results and the smoothness of the predictions have not been fully studied. It would be worthwhile to introduce multi-frame data for long-range 3D lane detection and tracking in future works.

6. Acknowledgement

This work was supported by the National Natural Science Foundation of China (61976170, 91648121), and Tencent Joint Research Program.

References

[1] Min Bai, Gellert Mattyus, Namdar Homayounfar, Shenlong Wang, Shrinidhi Kowshika Lakshmikanth, and Raquel Urtasun. Deep multi-sensor lane detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3102–3109. IEEE, 2018.

[2] Luca Caltagirone, Samuel Scheidegger, Lennart Svensson, and Mattias Wahde. Fast lidar-based road detection using fully convolutional neural networks. In *2017 ieee intelligent vehicles symposium (iv)*, pages 1019–1024. IEEE, 2017.

[3] Kuo-Yu Chiu and Sheng-Fuu Lin. Lane detection using color-based segmentation. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 706–711. IEEE, 2005.

[4] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022.

[5] Noa Garnett, Rafi Cohen, Tomer Pe’er, Roee Lahav, and Dan Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2921–2930, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Danilo Caceres Hernandez, Van-Dung Hoang, and Kang-Hyun Jo. Lane surface identification based on reflectance using laser range finder. In *2014 IEEE/SICE International Symposium on System Integration*, pages 621–625. IEEE, 2014.

[8] Namdar Homayounfar, Wei-Chiu Ma, Shrinidhi Kowshika Lakshmikanth, and Raquel Urtasun. Hierarchical recurrent attention networks for structured online maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3417–3426, 2018.

[9] Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. Dagmapper: Learning to map by discovering lane topology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2920, 2019.

[10] Yeongmin Ko, Younkwan Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] Chanho Lee and Ji-Hyun Moon. Robust lane detection and tracking for real-time applications. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):4043–4048, 2018.

[12] Minhyeok Lee, Junhyeop Lee, Dogyoon Lee, Woojin Kim, Sangwon Hwang, and Sangyoun Lee. Robust lane detection via expanded self attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 533–542, 2022.

[13] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258, 2019.

[14] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: a top-to-down lane detection framework based on conditional convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3773–3782, 2021.

[15] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021.

[16] Sheng Luo, Xiaoqin Zhang, Jie Hu, and Jinghua Xu. Multiple lane detection via combining complementary structural constraints. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7597–7606, 2020.

[17] Philipp Martinek, Gheorghe Pucea, Qing Rao, and Udhayaraj Sivalingam. Lidar-based deep neural network for reference lane generation. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 89–94. IEEE, 2020.

[18] Annika Meyer, Philipp Skudlik, Jan-Hendrik Pauls, and Christoph Stiller. Yolino: Generic single shot polyline detection in real time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2916–2925, 2021.

[19] Sandipann P Narote, Pradnya N Bhujbal, Abbhilasha S Narote, and Dhiraj M Dhane. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 73:216–234, 2018.

[20] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018.

[21] Dong-Hee Paek, Seung-Hyung Kong, and Kevin Tirta Wijaya. 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*, pages 4450–4459, 2022.

[22] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[23] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *European Conference on Computer Vision*, pages 276–291. Springer, 2020.

[24] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on local: Detecting lane marker from bottom up via key point. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14122–14130, 2021.

[25] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[26] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. 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*, pages 294–302, 2021.

[27] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021.

[28] Shenlong Wang, Min Bai, Gellert Mattyus, Hang Chu, Wenjie Luo, Bin Yang, Justin Liang, Joel Cheverie, Sanja Fidler, and Raquel Urtasun. Torontocity: Seeing the world with a million eyes. *arXiv preprint arXiv:1612.00423*, 2016.

[29] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *Advances in Neural Information Processing Systems*, 34:18381–18394, 2021.

[30] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3547–3554, 2021.