Spatio-Temporal Fusion-based Monocular 3D Lane Detection

Yin Wang[1*]
yinwang20@mails.jlu.edu.cn
Qiuyi Guo[2]
guoqiuyi@senseauto.com
Peiwen Lin[2]
linpeiwen@senseauto.com
Guangliang Cheng[2]
guangliangcheng2014@gmail.com
Jian Wu[1]
wujian@jlu.edu.cn

[1] Jilin University
Changchun, China
[2] SenseTime Research

Abstract

The monocular 3D lane detection (Lane3D) methods are increasingly proposed to address the issue of inaccurate bird-eye-view (BEV) results in various complex scenarios (*e.g.* up and downhills, bumps). However, there are a few restrictions on existing Lane3D methods. Primarily, only single-frame input is considered, which leads to poor results in no visual cues scenarios (*e.g.* obscured by surrounding vehicles). The other is that they rely on the camera pose to the road surface to translate to the road coordinate system. To address these issues and better exploit the spatio-temporal continuity of the lanes, we propose a novel Spatial-Temporal Lane3D model abbreviated as STLane3D. First, we propose a novel multi-frame pre-alignment layer under BEV, which uniformly projects features from different frames onto the same ROI region. Afterward, we propose a plug-and-play spatio-temporal attention module and a new 3DLane IOULoss. Experiments on the ONCE and OpenLane datasets demonstrate that our single-frame model, independent of camera extrinsic, can achieve close detection accuracy compared to the current state-of-the-art. And our multi-frame model improves the F1 score by 3.5% compared to the single-frame model on the ONCE dataset, which demonstrates the effectiveness of the multi-frame fusions strategy. Moreover, with multi-frame information, our model achieves satisfying performance in complex scenes lacking enough visual information and meets the real-time requirements on autonomous vehicles.

1 Introduction

Lane detection is a classic yet challenging computer vision task and an intuitive and effective way for autonomous vehicles to perceive their surroundings. It is widely used in autonomous driving scenarios (*e.g.* lane keeping, automatic cruise control, assisted trajectory planning),

* This work is done when Yin Wang is an intern at SenseTime Research.

where it can help self-driving cars better position themselves and improve safety. Traditional 2D lane detection methods focus on obtaining the exact lane position on the image. However, due to the lack of image depth information, the 2D results are challenging to apply to downstream tasks directly. We have to project them back into 3D space through inverse perspective mapping (IPM), which is under the assumption that the road surface is flat. Moreover, vehicle bumps and uneven road surfaces will significantly affect position accuracy. Monocular 3D lane detection (Lane3D) methods [3, 4, 5, 9, 10] are proposed to solve these issues, which can obtain the lane position from the 3D world space in an end-to-end manner without road planar assumption.

Although existing Lane3D methods have shown the ability to alleviate the weaknesses of 2D lane detection, most of them are carried out in the road coordinate system to remedy different camera mounting positions. The real-time pitch angle and camera height are needed during the training process. However, in practice, getting the real-time ground truth from the onboard sensors is challenging because the assumption of flat ground is not always satisfied. For example, in the scene of up and downhill, the ego pose relative to the global coordinate system is inconsistent with the road coordinate system due to the bumps of the ego car, which will introduce additional translation errors, as is shown in Fig. 1(c). On top of that, effort in solving the complex scenes with no visual cues (*e.g.* close dashed lines, lane lines obscured by surrounding vehicles) has not been well explored, which may introduce false negatives due to the lack of enough visual information.

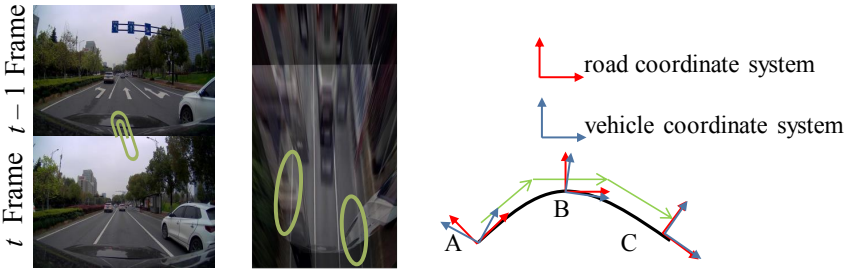
Based on the above observations, we aim to provide a simple and spatio-temporal fusion-based 3D lane detection method abbreviated as STLane3D to tackle these issues. Firstly, to eliminate the performance drop due to the insufficient visual cues, we propose a **multi-frame fusion mechanism** by making use of the strong spatio-temporal continuity in consecutive frames instead of focusing on single frame. Specifically, we utilize a pre-alignment operation to align the space information on the different frames. Then an attention module is introduced to fuse BEV features from different times. With this design, we can achieve a more complete and continuous perception of the lane features. For example, some parts of lanes are invisible in frame $t-1$ or frame t due to the occlusion of surrounding vehicles in Fig .1(a). By fusing multiple frames, the position information can be recognized in Fig .1(b), marked by the green circles. Secondly, considering the difficulty of obtaining a precise camera pose and recognizing the change of camera pose on the consecutive frames during the multi-frame fusion, we perform 3D lane detection under the camera coordinate system.

We conduct extensive experiments on the standard ONCE and OpenLane benchmarks. Compared to other Lane3D methods, our single-frame and multi-frame method achieve the state-of-the-art result with an F1 score of 77.53% on ONCE and 50.55% on OpenLane. To sum up, our main contributions are as follows:

(1) We present a novel, simple and effective Lane3D framework by incorporating both spatial and temporal information among frames. To the best of our knowledge, it is the first spatio-temporal 3D lane detection algorithm.

(2) We design a pre-alignment operation and plug-and-play spatio-temporal attention module, which can fuse multi-frame features more effectively.

(3) We propose 3DLane IOULoss, which constrains the lateral and vertical variation range of a lane and improves the performance of the model.



(a) (b) (c)

Figure 1: (a) the original picture of frame $t-1$ and frame t in same sequence, (b) the two frames after alignment under BEV, (c) non-coincidence of the vehicle coordinate system and the road coordinate system due to road bumps.

2 Related Work

2.1 3D Lane Detection

The 2D lane detection methods focus mainly on improving the detection performance on 2D images. However, the accuracy of lane detection under BEV is more concerned in downstream applications, such as the autonomous driving scenario. There are two main ideas to solve this problem:

Lidar fusion methods [1, 2, 11] integrate the accurate position information of Lidar and the color information of the camera. These methods achieve a significant improvement, especially for road edge detection.

Monocular 3D methods [3, 4, 5] transform the image to the road coordinate system through IPM transformation and predict the position on the road projection plane and the height of the road surface at the corresponding position simultaneously. Based on 3DLaneNet [4], GenLaneNet [5] corrects the projection relationship of the 3D lane and decouples the prediction of position and height into two stages. PersFormer [3] replaces the IPM layer with a deformable transformer to obtain enhanced BEV features. SALAD [10] predicts the position of the lane in the image and the corresponding inverse depth directly in the camera coordinate system. CLGO [9] adopts a 3D extended version of the curve parameter prediction method to predict the curve parameters of a 3D curve directly.

2.2 View-transformation and BEV temporal fusion

There are currently mainly three ways to transform the original image to BEV. The first is to use the IPM [4] based on the position of the camera related to the road surface. The second is to directly predict the depth of each point on the original image [6, 10]. The third is to do the transformation implicitly (*e.g.* using MLP layers [7] or a deformable transformer [8]), which inputs the original image and outputs the BEV. The latter two have the problem that it is difficult to decouple the model from camera intrinsic and extrinsic, so we use the first transformation method and convert the original image to a virtual BEV plane uniformly.

The multi-frame fusion under BEV [6, 8] has proven its effectiveness in 3D object detection. BEVFormer [8] fuses temporal information by spatial cross-attention and temporal self-attention, while BEVDet4D [6] stitches among BEV features from adjacent frames by concatenation operation.

3 STLane3D

In this section, we first formulate Lane3D Detection task in Sec. 3.1 and then present the overall pipeline of STLane3D in Sec. 3.2. Subsequently, we introduce our key contributions in detail, including the pre-alignment operation in Sec. 3.3.1, the spatio-temporal attention model in Sec. 3.3.2 and the 3DLane IOULoss in Sec. 3.3.3.

3.1 Problem Definition

Let $\mathbf{X^m} = \left[X^1, X^2, \ldots, X^t, \ldots, X^m\right]$ represents a video sequence of length m, and X^t denotes the t-th frame. We aim to predict a sequence of lanes $\mathbf{P}^m = \left[P^1, P^2, \ldots, P^t, \ldots, P^m\right]$, where P^t represents the lane positions of the t-th frame in the sequence. For each frame, $P^t = \left\{l^{0,t}, l^{1,t}, \ldots, l^{k,t}, \ldots, l^{C^t,t}\right\}$, where C^t is the number of lanes in t-th frame and $l^{k,t}$ is represented as a collection of 3D points as follows.

$$l^{k,t} = \left[(x_0^{k,t}, y_0^{k,t}, z_0^{k,t}), (x_1^{k,t}, y_1^{k,t}, z_1^{k,t}), ..., (x_{n_k}^{k,t}, y_{n_k}^{k,t}, z_{n_k}^{k,t})\right], \tag{1}$$

where k is the index of the lane and n_k is the max number of visible points of the k-th lane. As shown in Fig. 2, the coordinate system is defined as the camera coordinate system. The x-axis points to the right of the camera, the y-axis points to the front of the camera, and the z-axis points to the sky.

Due to the sparsity of 3D lanes in space, there is currently no dense representation similar to the 2D lane segmentation task. To keep the output of each frame with the same shape, we adopt the 3D lane anchor representation [3]. The anchors are lateral along x-axis, with an incline angle φ and a start point x_{start}. We set 16 start points and 7 incline angles for a total of $n_{\text{anchor}} = 112$ preset anchors. The 3D lane anchor representation $\ell^{k,t}$ for each frame at time t is represented in Fig. 2.

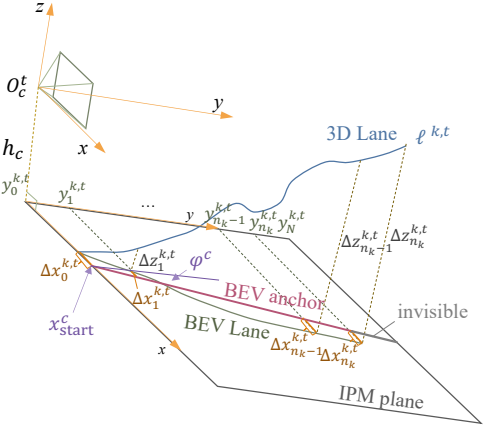


Figure 2: The camera coordinate system, whose origin is located at the camera optical center at time t. 3D lane $\ell^{k,t}$ represents the k-th lane at time t, corresponding to c-th BEV anchor with an incline angle φ^c and a start point x_{start}^c.

$$\ell^{k,t} = \left[(\Delta x_0^{k,t}, y_0^{k,t}, z_0^{k,t}, vis_0^{k,t}), \ldots, (\Delta x_i^{k,t}, y_i^{k,t}, z_i^{k,t}, vis_i^{k,t}), \ldots, (\Delta x_N^{k,t}, y_N^{k,t}, z_N^{k,t}, vis_N^{k,t})\right], \tag{2}$$

where $\Delta x_i^{k,t}$ and $vis_i^{k,t}$ represent the x offset and visibility of the i-th sampled point of the k-th lane in the t-th frame, N represents the number of points sampled uniformly along the y-axis for each anchor. After post-processing, we can obtain the final point set expression $l^{k,t}$. Similar to GenLaneNet [5], We present a position correction based on the analogous triangle in camera coordinate system in the appendix A.2.

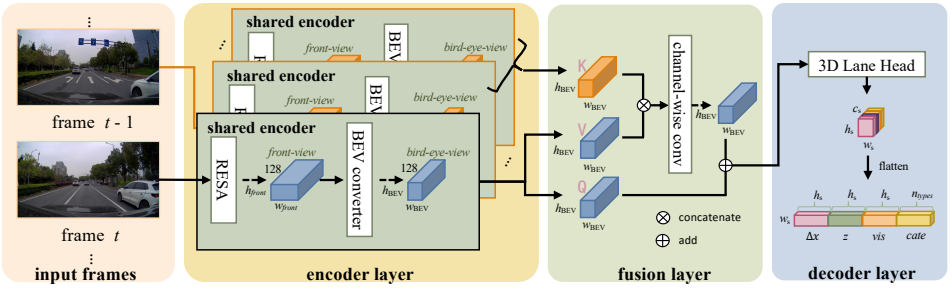


Figure 3: The overall paradigm of STLane3D. STLane3D is mainly divided into the following three main parts: (1) encoder layer, (2) fusion layer, and (3) decoder layer.

3.2 Overview

The overall paradigm of our method is shown in Fig. 3. The pipeline is mainly divided into three blocks: encoder layer, fusion layer, and decoder layer. It should be noted that the fusion layer used in multi-frame model is a plug-and-play module, while the single-frame model only consists of the encoder and decoder layers.

Encoder layer We adopt ResNext-50 combined with RESA block [12] as backbone, which can capture spatial relationships of pixels across rows and columns. The BEV converter projects the extracted features to the BEV plane through bilinear interpolation. The encoder is shared at different timestamps and can reduce inference time by saving the feature map of historical frames to alleviate redundant computation.

Fusion layer We use channel-wise convolution to fuse the information of historical and current frames. We take the features from history and current frames as key-value pairs, respectively, and the features from the current frame as the query. We use the ego-motion from RTK to pre-align the features of the historical frame before fusion, which can ensure the position correspondence during 1×1 convolution. BEV pre-alignment will be introduced in detail in Sec. 3.3.1.

Decoder layer We decode the feature by the 3D lane head through a series of convolutions with no padding in the y dimension. It will be finally flattened into the prediction result of $(w_s, 3 \times h_s + n_{\text{types}})$, where w_s and h_s respectively represent the number of anchors we set in the width direction of the image and the number of sampled points in the height direction of the image, n_{types} represents the prediction of the lane category. For each preset anchor, we predict, (1) **lateral position offset** Δx relative to the anchor, (2) **height** z relative to the virtual plane, (3) **visibility** vis of the sampled point, (4) **category** $type$ of the lane.

3.3 Model details

3.3.1 BEV pre-alignment

As described in Sec. 1, front-view perspective fusion will bring many difficulties. However, it also raises two problems when transforming the view from the original image to the BEV. One is **how to cast** the original image onto the BEV, and the other is **how to align** BEV features from two frames. It is common practice to do these two steps separately, first projecting each frame onto BEV and then aligning them [6, 8]. We minimize the conversion error by

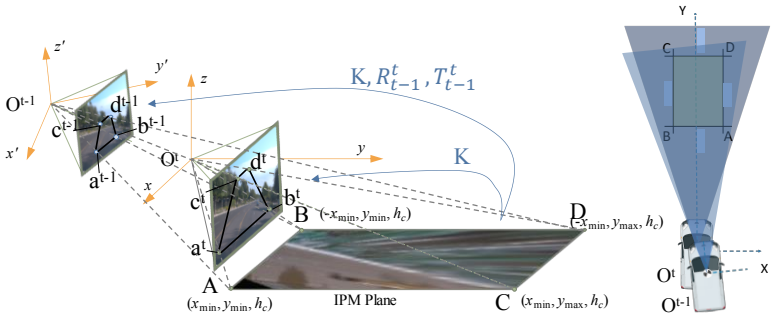


Figure 4: BEV pre-alignment. As shown in (b), the ego vehicle moves from O_c^{t-1} to O_c^t, and the corresponding pose transformation is R^t_{t-1}, T^t_{t-1}. The green rectangle $ABCD$, corresponding to the IPM Plane in (a), represents the BEV ROI of t frame.

combining these two steps into single image feature processing.

By analyzing the process of IPM, the core operation is to find the region of interest (ROI) layout in the original image. We assume that our ROI (the area enclosed by the four points of A,B,C,D as shown in Fig. 4) is $x \in [x_{\min}, x_{\max}]$ and $y \in [y_{\min}, y_{\max}]$ under camera coordination and the our camera is mounted at a virtual height h_c. The corresponding points a^t, b^t, c^t, d^t on the original image are computed according to the pinhole camera model:

$$(u^t, v^t, 1)^\top = K_{3\times4} \cdot (x^t, y^t, z^t, 1)^\top \tag{3}$$

where u^t, v^t is coordinates in the image plane expressed in pixel units, x^t, y^t, z^t is a 3D point defined in the camera coordinates system at t frame, $K_{3\times4}$ is the camera intrinsic. Then the area enclosed by a^t, b^t, c^t, d^t on the original image is stretched into the BEV plane in the shape of $(h_{\text{BEV}}, w_{\text{BEV}})$ through bilinear interpolation.

According to the RTK motion information of the ego-vehicle, we can get the camera pose transformation from $t-1$ frame to t frame with R^t_{t-1}, T^t_{t-1}. Therefore, we can easily obtain the coordinates of A,B,C,D in the camera coordinate system of the $t-1$ frame, using the pose transformation formula defined as

$$R^t_{t-1} \cdot [x^{t-1}, y^{t-1}, z^{t-1}, 1]^\top + T^t_{t-1} = [x^t, y^t, z^t, 1]^\top. \tag{4}$$

Also according to the pinhole camera model, we can get the image points $a^{t-1}, b^{t-1}, c^{t-1}, d^{t-1}$ corresponding to the $t-1$ frame.

3.3.2 Attention Module

Since we perform the BEV pre-alignment operation before fusion, the features from different frames has one-to-one correspondence in image locations. We use the features from $t-1$ frame as the **key**, and the features from t frame as the **value** and **query** to build a simple and effective attention module to fuse features from different moments. We fuse the **key** and **value** with simple channel-wise convolutions. We present detailed description of attention module in the appendix A.1.

3.3.3 3DLane IOULoss

LaneIOU Loss is firstly proposed by CLRNet [13] applied to the Lane2D task to regress the lane prior as a whole unit and tailor for Lane2D dataset evaluation metrics. We introduce the LaneIOU Loss to Lane3D task and modified it to better fit the Lane3D task.

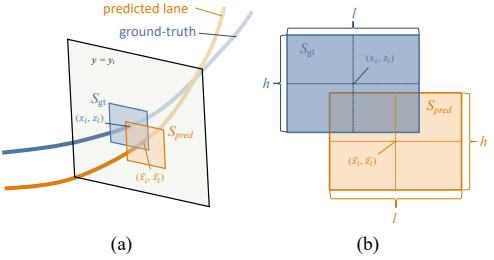


Figure 5: 3DLane IOULoss.

3DLane IOULoss extends LaneIOU Loss from 2D space to 3D space. Compared with L1 Loss, it can maintain the continuity of lanes more effectively as it couples the errors in the lateral and height directions. The ground-truth labels $(x_i, z_i), i = 0, 1, ..., N$ at the position of $y = y_i$ are obtained by interpolation. As shown in Fig. 5, two rectangles with a length of l and a height of h are denoted as the ground-truth result S_{gt} and the predicted result S_{pred}. 3DLane IOULoss can be expressed as their intersection-over-union (IOU):

$$L_{\text{reg}} = \frac{S_{\text{gt}} \bigcap S_{\text{pred}}}{S_{\text{gt}} \bigcup S_{\text{pred}}}. \tag{5}$$

Besides the L_{reg}, we use the binary cross-entropy loss L_{vis} and L_{prob} follow [5]. For the input image, the total loss can be expressed as,

$$L_{\text{tot}} = \sum_{k}^{n_{\text{anchor}}} \sum_{i}^{N} (\alpha \cdot L_{\text{prob}}^{k} + \beta \cdot p^{k} \cdot L_{\text{vis}}^{i,k} + \gamma \cdot p^{k} \cdot L_{\text{reg}}^{i,k}), \tag{6}$$

where p^k represents whether the k-th anchor contains ground-truth lane, n_{anchor} is the number of anchors, α, β and γ are three hyperparameters.

4 Experiments

4.1 Experimental Setting

Dataset. To extensively evaluate the proposed method, we conduct experiments on ONCE [10] and OpenLane [3]. They are both large-scale autonomous driving datasets with 3D lane annotations. OpenLane is the first real-world dataset for 3D lane detection and ONCE is the up-to-date largest real-world dataset for 3D lane detection including 1 million scenes and 211k 3D lane fully annotated scenes.

Metrics. We adopt the evaluation metrics of ONCE [10] on ONCE dataset. The metric first calculate the matching degree of two lanes on the z-x plane. To be concrete, lane is represented as $L^k = \{(x_i^k, y_i^k, z_i^k)\}_{i=1}^{n}$. The curve matching error $CD_{p,g}$ between L^p and L^g is calculated as follows:

$$\begin{cases} CD_{p,g} = \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} ||P_{g_i} - \hat{P}_{p_j}||_2 \\ \hat{P}_{p_j} = \underset{P_{p_j} \in L^p}{min} ||P_{p_j} - P_{g_i}||_2 \end{cases} \tag{7}$$

We use F1-measure as the evaluation metrics on OpenLane dataset.

4.2 Implementation

Training. Following GenLaneNet [5], we train STLane3D in a two-stage manner to alleviate unnecessary extra computation. In the first stage, we train the 2D lane model using 2D lane labels on the ONCE dataset for 50 epochs. In the second stage, we retain the weights of the previously pre-trained encoding layers and use the 3D lane labels to train single-frame and multi-frame models for 8 epochs, respectively.

The farthest distance of the annotated lane in the ONCE dataset is 50 meters in front of the vehicle. And the average forward distance between the two frames is 5 meters. So we set the distance range of the BEV projection to [-100m, 100m] of the vehicle and sample 72 points at equal intervals to form the anchors. The rectangular window of 3DLane IOULoss is set to $l = 3$m, $h = 1$m. The overall loss weight is set to $\alpha = 10, \beta = 10, \gamma = 1$. We use an Adam optimizer and use a fixed learning rate of 0.0001.

Inference speed. The inference speed is the average upon validation and test splits using ResNext-50 as the backbone. The inference speed of our single-frame-model is 63 FPS on a single RTX 3060, and our multi-frame-model is 30 FPS(n_{frames} = 4) by setting the batch size to 1.

4.3 Main Results

As shown in Tab. 1, we compare the proposed STLane3D with 3DLaneNet [4], GenLaneNet [5], SALAD [10] and PersFormer [3]. Compared with the single-frame model, the F1-score of the multi-frame model increases by 3.5%, and compared with PersFormer, it increases by 3.2%, and achieved the new state-of-art F1-score of 77.53%.

Table 1: Comparison with the state-of-the-art methods on the ONCE dataset.

Methods	F1-score(%)$_{\uparrow}$	Precision(%)$_{\uparrow}$	Recall(%)$_{\uparrow}$	CD-error(m)$_{\downarrow}$
3DLaneNet [4]	44.73	61.46	35.16	0.127
GenLaneNet [5]	45.59	63.95	35.42	0.121
SALAD [10]	64.07	75.90	55.42	0.098
PersFormer [3]	72.07	77.82	67.11	0.086
STLane3D-single(ours)	74.05	76.63	71.64	0.085
STLane3D-multi(ours)	**77.53**	**81.54**	**73.90**	**0.066**

4.4 Ablation Studies and Discussions

We present ablation study results to demonstrate the effectiveness of each component in STLane3D and further discuss the reasons for the ups and downs of performance.

Effectiveness of the Multi-frame Fusion We investigate the effectiveness of the multi-frame fusion pre-alignment operation. Detailed ablation results are presented in Fig. 7. The experimental results show that the accuracy of the model with pre-alignment operation first increases and then decreases as the number of fused frames increases. The highest accuracy is achieved when $n_{\text{frames}} = 4$. The accuracy of the model without pre-alignment decreases gradually as the n_{frames} increase. This suggests that the pre-alignment operation facilitates the fusion of multiple frames, while introducing too old historical frames can also impact the model.

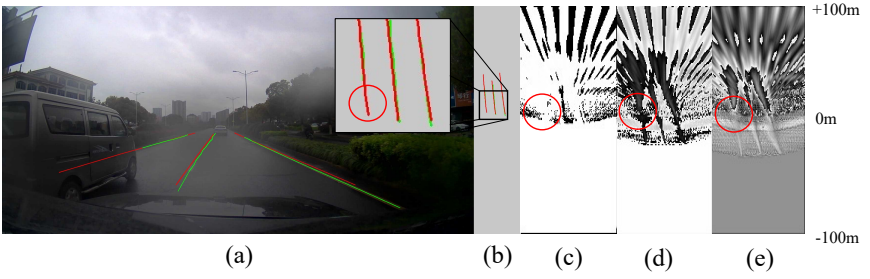


Figure 6: Illustration of an obscured scene.

Fig. 6 shows a scene where the lane line is obscured by a vehicle on the left. Part of the lane line is not visible in current frame due to the occlusion of the left car, while this lane is observable from history frames. Thus, we can use the history information (with multi-frames) to solve cases with few visual cues and also provide the information on the lane lines behind the ego car (< 0m), which helps to improve the accuracy of the close lane position. Fig. 6 (c), (d), (e) are the results for the single frame, 4 frames, and 6 frames models, respectively. (a) and (b) show the visualization of the predicted result (*Red*) and the label (*Green*) in the original and BEV images. The first output channel of the fusion layer is shown for $n_{\text{frames}} = 4$ and $n_{\text{frames}} = 6$ respectively in (d) and (e), and the output channel of the encoder layer of the single-frame model is shown in (c).

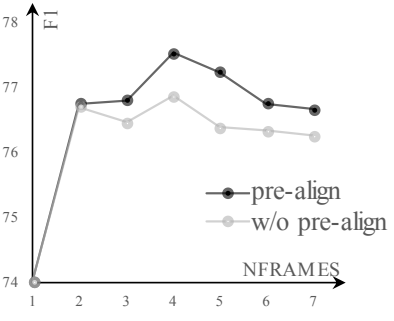


Figure 7: Effectiveness of the multi-frame fusion.

Effectiveness of the 3DLane IOULoss We investigate the effectiveness of 3DLane IOULoss, including the comparison of L1 Loss, LaneIOU Loss [13] and 3DLane IOULoss (ours). Detailed ablation results are presented in Tab. 2.

3DLane IOULoss improves the F1-score by 1.55% and CD-error by 0.005m, compared with the L1 Loss function, demonstrating the effectiveness of 3DLane IOULoss. At the same time, we found that during the training process, using the L1 Loss, the loss function is occasionally infinite. This is due to the lack of an upper limit when using the L1 Loss, and the upper limit of the IOU-based loss function is 1, so the above problems will not occur.

Table 2: Effectiveness of the 3DLane IOULoss.

Methods	F1-score(%)↑	Precision(%)↑	Recall(%)↑	CD-error(m)↓
L1 Loss	75.98	76.99	**74.99**	0.071
LaneIOU Loss	76.85	79.75	74.15	0.068
3DLane IOULoss(ours)	**77.53**	**81.54**	73.90	**0.066**

Experiment results on OpenLane We tested our model on OpenLane as well. Although the camera intrinsic and extrinsic vary considerably between sequences of OpenLane compared to ONCE, we use the same parameter settings as on ONCE and use a uniform BEV plane height of 1.5m. Due to the variations in pitch angle (camera mounting) among sequences on OpenLane, there is an ROI mismatch in front of the ego car. A similar problem

exists in the lateral direction as well, which significantly affects the performance of uphill & downhill and curve cases. Meanwhile, we sampled the frames at 4 frame intervals on OpenLane to maintain consistency with ONCE. We achieved close results with state-of-the-art methods, demonstrating that it is feasible to predict directly in the camera coordinate system independent of the camera pose. Experimental results show that STLane3D performs well in scenes without visual cues, such as at night, in extreme weather, etc. Qualitative results will be illustrated in appendix A.3. The specific experimental results are shown in Tab. 3.

Table 3: Comparison with the state-of-the-art methods on the OpenLane dataset. We report F1-score for different scenarios and overall F1-score(%), X-error(m) and Z-error(m).

Methods	Up & Down	Curve	Extreme Weather	Night	Inter-section	Merge & Split	F1-score	X-error	Z-error
3DLaneNet [4]	40.8	46.5	47.5	41.5	32.1	41.7	44.1	0.572	0.443
GenLaneNet [5]	25.4	33.5	28.1	18.7	21.4	31.0	32.3	0.684	0.521
PersFormer [3]	**42.4**	**55.6**	48.6	46.6	40.0	**50.7**	50.5	0.553	0.431
STLane3D(ours)	41.3	47.4	**54.0**	**51.3**	**42.5**	47.9	**50.6**	**0.500**	**0.178**

5 Conclusion

In this paper, to address the lack of temporal information and dependence on camera poses, which are common problems of existing Lane3D methods, we present a novel, simple and effective 3D lane detection framework STLane3D by incorporating both spatial and temporal information among frames. We propose a pre-alignment operation, a plug-and-play spatio-temporal attention module, and a novel 3DLane IOULoss. Experiments on ONCE and OpenLane demonstrate the effectiveness of the multi-frame fusion strategy. STLane3D achieves state-of-the-art detection accuracy and performs well in complex scenarios lacking sufficient visual information.

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, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[3] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision (ECCV)*, 2022.

[4] 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.

[5] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *European Conference on Computer Vision*, pages 666–681. Springer, 2020.

[6] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022.

[7] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework, 2021.

[8] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers.

[9] Ruijin Liu, Dapeng Chen, Tie Liu, Zhiliang Xiong, and Zejian Yuan. Learning to predict 3d lane shape and camera pose from a single image via geometry constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1765–1772, 2022.

[10] Fan Yan, Ming Nie, Xinyue Cai, Jianhua Han, Hang Xu, Zhen Yang, Chaoqiang Ye, Yanwei Fu, Michael Bi Mi, and Li Zhang. Once-3dlanes: Building monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17143–17152, 2022.

[11] Xinyu Zhang, Zhiwei Li, Xin Gao, Dafeng Jin, and Jun Li. Channel attention in lidar-camera fusion for lane line segmentation. *Pattern Recognition*, 118:108020, 2021.

[12] 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.

[13] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 898–907, June 2022.