Learning to Predict 3D Lane Shape and Camera Pose from a Single Image via Geometry Constraints

Ruijin Liu, [1] **Dapeng Chen,** [2] **Tie Liu,** [3] **Zhiliang Xiong,** [4] **Zejian Yuan** [1]
[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[2] The Hong Kong Polytechnic University, Hong Kong SAR, China
[3] College of Information Engineering, Capital Normal University, Beijing, China
[4] Shenzhen Forward Innovation Digital Technology Co. Ltd, Shenzhen, China
lrj466097290@stu.xjtu.edu.cn, dapengchenxjtu@yahoo.com, liutiel@163.com, leslie.xiong@forward-innovation.com, yuan.ze.jian@xjtu.edu.cn

Abstract

Detecting 3D lanes from the camera is a rising problem for autonomous vehicles. In this task, the correct camera pose is the key to generating accurate lanes, which can transform an image from perspective-view to the top-view. With this transformation, we can get rid of the perspective effects so that 3D lanes would look similar and can accurately be fitted by low-order polynomials. However, mainstream 3D lane detectors rely on perfect camera poses provided by other sensors, which is expensive and encounters multi-sensor calibration issues. To overcome this problem, we propose to predict 3D lanes by estimating camera pose from a single image with a two-stage framework. The first stage aims at the camera pose task from perspective-view images. To improve pose estimation, we introduce an auxiliary 3D lane task and geometry constraints to benefit from multi-task learning, which enhances consistencies between 3D and 2D, as well as compatibility in the above two tasks. The second stage targets the 3D lane task. It uses previously estimated pose to generate top-view images containing distance-invariant lane appearances for predicting accurate 3D lanes. Experiments demonstrate that, without ground truth camera pose, our method outperforms the state-of-the-art perfect-camera-pose-based methods and has the fewest parameters and computations. Codes are available at https://github.com/liuruijin17/CLGo.

Introduction

Compared with 2D lane detection, image-based 3D lane detection is beneficial to perceiving a real-world driving environment, which is crucial for intelligent cruise control, high definition map construction, and traffic accident reduction in autonomous driving (Homayounfar et al. 2018; Efrat et al. 2020). Unlike the 2D lane detection that relies on the flat ground plane assumption, the 3D method is more flexible to handle complex road undulations. It usually requires a camera pose to transform the image/features of the perspective view to the top-view by inverse perspective mapping. The current methods utilize the ground truth camera pose provided by the benchmark to estimate the accurate top-view feature. Such a strategy is potentially expensive in realistic driving applications because it needs an additional third-party tool (inertial sensor or SfM algorithm (Clark et al. 2017; Brahmbhatt et al. 2018)) to provide an accurate camera pose steadily during driving (pandiii 2021; Guo et al. 2020)

Instead of utilizing the ground truth camera pose, we propose to learn camera pose online to perform view transform for 3D lane detection and impose geometry consistency constraints to improve the pose results. Specifically, we utilize a two-stage framework to learn from different viewpoints. The first stage aims at learning camera pose from perspective-view images. To improve the camera pose, we introduce geometry constraints. The constraints are based on an auxiliary network branch that predicts the 3D lanes. With the predicted camera pose, the predicted 3D lanes are projected to the camera plane and the virtual flat ground plane, which can be supervised by the ground truth lanes of the two planes. The second stage uses transformed top-view images based on learned camera pose to detect ultimate 3D lanes. In the top-view, lanes have a similar appearance along a longitudinal direction which is crucial for deciphering 3D lanes, especially in distant areas. The 3D lanes are modeled by two polynomials: one approximates the lateral variations, and the other expresses undulation changes along longitudinal positions. To better capture lanes' long and thin structures, both stages utilize transformers (Vaswani et al. 2017) to aggregate non-local context.

Our method is evaluated on the only public synthetic 3D lane detection benchmark and a self-collected real-world dataset which will be released. Without utilizing the ground truth camera poses, our method outperforms previous state-of-the-art methods that need ground truth camera poses. Furthermore, our approach has a light model size, few computation costs, and quite fast speed, showing great potential in real driving applications. The main contributions are summarized as follows:

- We design a two-stage framework that firstly predicts the camera pose, then utilizes the camera pose to generate a top-view image for accurate 3D lane detection.
- We propose geometry constraints to assist in estimating camera pose, which enforces the consistency between the predicted 3D poses and lanes with the ground truth lanes on the 2D plane.
- We employ polynomials to model the 3D lane, which preserves the lanes' local smoothness and global shape

and avoids the complex post-processing.

Related Work

The field of vision-based lane detection has grown considerably in the last decade. The popularity of camera sensors has allowed lane detection in 2D to gain significant momentum. Traditional methods typically design hand-crafted features, adopt mathematical optimized algorithms and geometry or scene context to learn lane structures well (Narote et al. 2018; Niu et al. 2016; Wang, Shen, and Teoh 2000). Deep methods built by convolutions have exploded in recent years, making significant progress and applicable systems for real applications (Chen et al. 2018; Zhao, Yuan, and Chen 2020). Two-stage methods which extract segmentation or proposal plus post-processing ruled the filed for several years (Neven et al. 2018; Philion 2019; Pan et al. 2018; Zhang et al. 2018; Hou et al. 2019; Ko et al. 2020; Li et al. 2020; Tabelini et al. 2021; Qin, Wang, and Li 2020; Xu et al. 2020; Jung et al. 2020; Yoo et al. 2020; Lee et al. 2021; Zheng et al. 2020). To streamline the pipeline into an end-to-end fashion, single-stage methods (Torres et al. 2020; Liu et al. 2021) directly estimate coefficients of prior mathematical curves have shown both higher efficacy and efficiency. However, those 2D detectors are built with specific planar world assumptions, resulting in a limited representation of realistic and complex 3D road undulations.

The 3D lane detection task has attracted research interest because it doesn't rely on the plane assumption to predict the lanes (Coulombeau and Laurgeau 2002; Benmansour et al. 2008; Bai et al. 2018; Garnett et al. 2019; Guo et al. 2020; Efrat et al. 2020; Jin et al. 2021). However, detecting a 3D lane from a single RGB image is non-trivial. The results can be easily affected by appearance variation due to changes in lighting condition, texture, background, and camera pose. One solution is to utilize the extra information from other sensors such as LiDAR or stereo cameras (Coulombeau and Laurgeau 2002; Benmansour et al. 2008; Bai et al. 2018). Still, the cost of these devices is too expensive to be applied to them on consumer-grade products widely (Guo et al. 2020), while arising multi-sensor calibration problems during driving. Unlike multi-sensor methods, monocular methods (Garnett et al. 2019; Guo et al. 2020; Efrat et al. 2020) only require a single image and camera pose to transform image/features from perspective-view to top-view for accurate 3D lane detection. Previous methods rely on the ground truth camera pose for evaluation and cannot work well when the camera pose changes dynamically when driving at rough terrain or accelerating (Zhou et al. 2021)). In contrast, our method provides a full-vision-based 3D lane detector that is not dependent on the ground truth camera pose and other sensors. It can flexibly adapt to the changing driving environment with an affordable device cost.

Methodology

3D Lane Geometry

We first define notations for the lane representation in the 3D space, perspective image plane, and a virtual flat ground plane as shown in Fig. 1 . Then we introduce the transformations between these spaces.

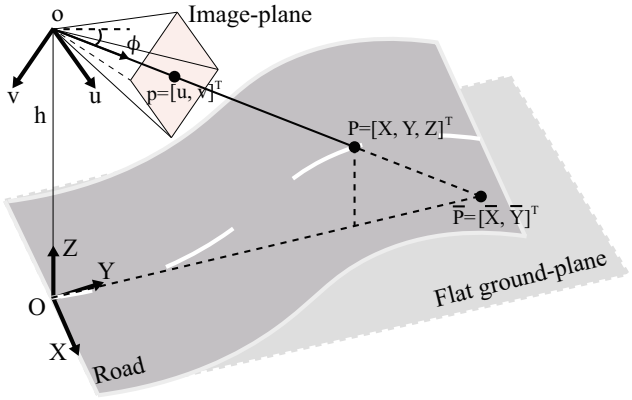

Figure 1: Geometry setup about camera and 3D lane.

Notation of Lane. Suppose $\mathbf{P} = [X, Y, Z]^T$ is a lane point in the 3D space. Its projection on the flat ground-plane (i.e., the plane with Z=0 in the 3D space) is denoted by $\bar{\mathbf{P}} = \left[\bar{X}, \bar{Y}\right]^T$, while its projection on the camera plane is denoted by $\mathbf{p} = [u, v]^T$. In the 3D space, the origin of the coordinate $\mathbf{O}$ is the perpendicular projection of the camera center $\mathbf{o}$ on the plane $Z = 0$. For the camera, we fix the intrinsic parameters f_x, f_y, c_x and c_y, and only estimate the camera height h and pitch angle ϕ following the common setup (Garnett et al. 2019; Guo et al. 2020).

Geometric Transformation. The geometric transformation projects points of the lanes from the 3D space to the flat ground plane. In particular, a point $\mathbf{P}$ in the 3D space should be on the same line with the camera center $\mathbf{o}$ and the projected point $\bar{\mathbf{P}}$ on the flat ground plane. Therefore, we have $\frac{h}{h-Z} = \frac{\bar{X}}{X} = \frac{\bar{Y}}{Y}$. Note that it holds no matter Z is positive or negative. The geometric transformation from 3D space to the flat ground-plane is written as:

$$\begin{bmatrix} \bar{X} \\ \bar{Y} \end{bmatrix} = \begin{bmatrix} \frac{h}{h-Z} & 0 & 0 \\ 0 & \frac{h}{h-Z} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{1}$$

For the simplicity of the notation, Eq. 1 is also formulated as $\bar{\mathbf{P}} = \mathbf{G}\mathbf{P}$, and $\mathbf{G}$ is the geometric transformation matrix.

Homography Transformation. The homography transformation projects the point from the flat ground plane to the image plane. Given a point $\bar{\mathbf{P}}$ on the flat ground plane and its corresponding point $\mathbf{p}$ on the image plane, the homography transformation are performed with their homogeneous coordinates:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\phi + \frac{\pi}{2}\right) & h \\ 0 & \sin\left(\phi + \frac{\pi}{2}\right) & 0 \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{Y} \\ 1 \end{bmatrix}, \tag{2}$$

We denote $\tilde{\mathbf{p}} = [\tilde{u}, \tilde{v}, \tilde{z}]^T$ and $\tilde{\mathbf{P}} = \left[\bar{X}, \bar{Y}, 1\right]^T$ as the homogeneous coordinates of $\mathbf{p}$ and $\bar{\mathbf{P}}$, therefore $u = \tilde{u}/\tilde{z}, v = \tilde{v}/\tilde{z}$. For simplicity, Eq. 2 is also formulated as $\tilde{\mathbf{p}} = \mathbf{H}\tilde{\mathbf{P}}$, and $\mathbf{H}$ is the Homography transformation matrix.

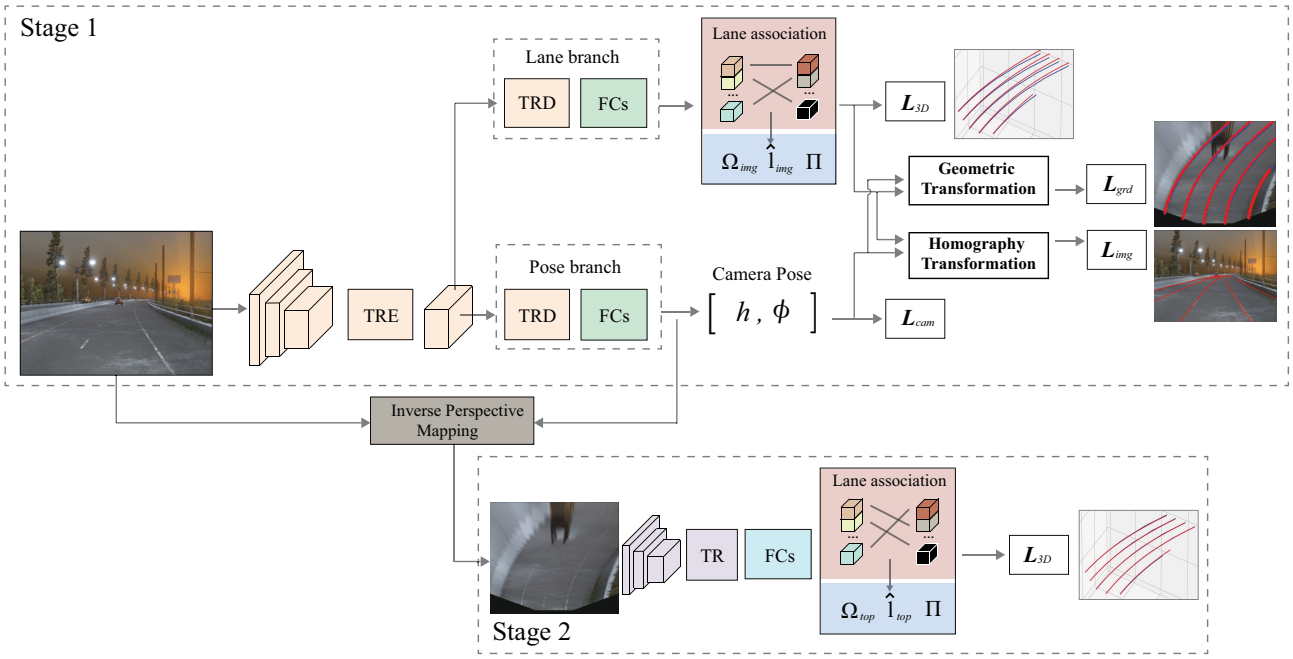

Figure 2: System Overview. Stage 1 learns camera pose with the help of an auxiliary lane branch and geometry constraints. Then, the estimated camera pose transforms image from perspective-view into top-view where lanes look similar. Finally, Stage 2 aims at predicting ultimate 3D lanes from distance-invariant top-view image accurately.

Output Definition

Our method outputs both the camera pose and the detected lane in the 3D space.

Camera Pose. As stated in the aforementioned geometry setting, the output camera pose contains the camera height h and camera pitch angle ϕ.

3D Lane. We use two polynomials to represent a 3D lane. For any point $[X, Y, Z]$ on the lane, X, Z can be represented as polynomial functions of Y, that is:

$$\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} \sum_{r=0}^{R} a_r Y^r \\ \sum_{r=0}^{R} b_r Y^r \end{bmatrix}, \tag{3}$$

where $a_R \neq 0$, $b_R \neq 0$ and a_r, b_r are real number coefficients. R indicates the order of the polynomial. Let t_{1n} and t_{2n} be the upper bound and lower bound of Y of the n-th lane, the n-th lane in an image can be represented by the coefficients of the polynomial functions and the bounds of Y:

$$\theta_n = \left\{ \{a_{rn}, b_{rn}\}_{r=0}^{R}, t_{1n}, t_{2n} \right\}, \tag{4}$$

where $n \in \{1, ..., N\}$ and N is the number of ground truth 3D lanes in the image.

Network Architecture

Fig. 2 illustrates the overall two-stage structures. Stage 1 contains a backbone for feature extraction over the entire image and a transformer-encoder (TRE) that aggregates non-local relations among spatial features. The output of TRE is fed to two branches: a pose branch that can decode ego camera pose and a lane branch for 3D lanes.

With an estimated camera pose, *inverse perspective mapping* (IPM) transforms the image into a top-view image. Then stage 2 employs a similar transformer-based network to be specialized at only predicting 3D lanes since lifting to top-view space makes the 3D lane fitting a much more simple task due to distance-invariant appearance features.

Stage 1. Given an image, stage 1 extracts convolution features, then aggregates spatial features with a backbone and transformer-encoder just like LSTR (Liu et al. 2021). After obtaining the encoded feature, the pose branch decodes the camera pose (camera height h and pitch angle ϕ) of the current image. To help the camera pose learning, we introduce a lane branch to decode 3D lanes. The 3D lanes are then projected to the flat ground plane and the image plane by the predicted camera pose, which are supervised by the ground truth lanes on the two planes.

Stage 2. Stage 2 mainly follows the pipeline in stage 1 with the only lane branch to estimate 3D lanes, which is the ultimate detection results for evaluation. Different from stage 1, stage 2 only adopts the transformed top-view images where lanes have a similar shape, scale and appearance to help reconstruct 3D lanes accurately.

Loss Functions with Geometry Constraints

We train the network of Stage 1 with geometry constraints by using $L_{s1} = L_{cam} + L_{3D} + L_{grd} + L_{img}$, where L_{3D} and L_{cam} are basic fitting losses to supervise the 3D lane and the camera pose, respectively. While L_{img} and L_{grd} serve as geometry constraints that assist the learning of camera pose. In stage 2, only L_{3D} is employed to train the network for

more accurate 3D lane detection. We now introduce these loss functions one by one.

Camera Pose Regression Loss. The regression loss for camera pose has the form of: $L_{cam} = \left|\hat{h} - h\right| + \left|\hat{\phi} - \phi\right|$, where $|\cdot|$ is the mean absolute error.

3D Lane Fitting Loss. The loss is used to supervise the lane in the 3D space. It is noteworthy that the lane branch outputs the polynomial coefficients of M 3D lanes, where M is larger than the maximum number of labeling lanes in the dataset. Because the loss function does not know the association between the predicted lanes and the ground truth lanes, we first associate the predicted curves and ground truth lanes by solving a bipartite matching problem.

3D Lane Association. Let $\Omega = \{\omega_m | \omega_m = (c_m, \theta_m)\}_{m=1}^{M}$ be the set of the predicted 3D lanes, where $c \in \{0, 1\}$ (0: none-lane, 1: lane), and $\theta_m = \left\{\{a_{rm}, b_{rm}\}_{r=0}^{R}, t_{1m}, t_{2m}\right\}$. Let $\Pi = \left\{\hat{\pi}_m | \hat{\pi}_m = \left(\hat{c}_m, \hat{\mathbf{L}}_m\right)\right\}_{m=1}^{M}$ be the set of ground truth 3D lanes, where the m-th ground truth lane $\hat{\mathbf{L}}_m = \left[\hat{X}_{km}, \hat{Y}_{km}, \hat{Z}_{km}\right]_{k=1}^{K}$. Note that Π is padded with non-lanes to fill enough the number of ground truth lanes to M for associating with Ω. Next, we form a bipartite matching problem between Ω and Π to build lane association. The problem is formulated as a distance minimizing task by searching an injective function $l : \Pi \to \Omega$, where $l(m)$ is the index of a 3D lane prediction $\omega_{l(m)}$ which is assigned to m-th ground truth 3D lane $\hat{\pi}_m$:

$$\hat{l} = \arg\min_{l} \sum_{m=1}^{M} D\left(\hat{\pi}_m, \omega_{l(m)}\right), \tag{5}$$

based on the matching cost:

$$\begin{aligned} D = &-\alpha_1 p_{l(m)}(\hat{c}_m) + \mathbb{1}(\hat{c}_m = 1)\,\alpha_2 \left|\hat{\mathbf{L}}_m - \mathbf{L}_{l(m)}\right| \\ &+ \mathbb{1}(\hat{c}_m = 1)\,\alpha_3 \left(\left|\hat{Y}_{1m} - t_{1l(m)}\right| + \left|\hat{Y}_{Km} - t_{2l(m)}\right|\right), \end{aligned} \tag{6}$$

where α_1, α_2 and α_3 are coefficients which adjust the loss effects of classification, polynomials fitting and boundaries regression, and $\mathbb{1}(\cdot)$ is an indicator function. The $l(m)$-th prediction lane $\mathbf{L}_{l(m)} = \left[X_{kl(m)}, Y_{kl(m)}, Z_{kl(m)}\right]_{k=1}^{K} = \left[\sum_{r=0}^{R} a_{rl(m)} \hat{Y}_{km}^{r},\ \hat{Y}_{km},\ \sum_{r=0}^{R} b_{rl(m)} \hat{Y}_{km}^{r}\right]_{k=1}^{K}$.

3D Lane Fitting. After getting the optimized injective function $\hat{l}$ by solving Eq. 5 using Hungarian algorithms (Carion et al. 2020), the 3D fitting loss can be defined as:

$$\begin{aligned} L_{3D} = &\sum_{m=1}^{M} \mathbb{1}(\hat{c}_m = 1)\,\alpha_3 \left(\left|\hat{Y}_{1m} - t_{1\hat{l}(m)}\right| + \left|\hat{Y}_{Km} - t_{2\hat{l}(m)}\right|\right) \\ &+ \mathbb{1}(\hat{c}_m = 1)\,\alpha_2 \left|\hat{\mathbf{L}}_m - \mathbf{L}_{\hat{l}(m)}\right| - \alpha_1 \log p_{\hat{l}(m)}(\hat{c}_m). \end{aligned} \tag{7}$$

where α_1, α_2 and α_3 are the same coefficients with Eq. 6.

Geometry Constraint in Flat Ground Plane. Projecting lanes to the flat ground plane provides global shape patterns correlated with varying lane heights. If lanes in 3D space have no undulation, their projections on this plan will show parallel alignment and the same curvature variation. However, when encountering an uphill road, lane projections will not be parallel but converging in the bottom (or diverging for a downhill road). Such shape patterns are clearer to guide the network to learn 3D lanes and camera heights in fine detail. With the same optimal $\hat{l}$, the fitting loss in the flat ground-plane can be written as:

$$L_{grd} = \sum_{m=1}^{M} \mathbb{1}(\hat{c}_m = 1)\,\alpha_2 \left|\hat{\bar{\mathbf{L}}}_m - \bar{\mathbf{L}}_{\hat{l}(m)}\right|. \tag{8}$$

Here, $\hat{\bar{\mathbf{L}}}_m = \left[\hat{\bar{X}}_{km}, \hat{\bar{Y}}_{km}\right]_{k=1}^{K}$ is the m-th ground truth projected lane. It is obtained by projecting the 3D ground truth lane by geometric transformation (Eq. 1), where the geometric matrix is calculated by the ground truth camera height $\hat{h}$. Similarly, we can obtain the $\hat{l}(m)$-th predicted projected lane $\bar{\mathbf{L}}_{\hat{l}(m)} = \left[\bar{X}_{k\hat{l}(m)}, \bar{Y}_{k\hat{l}(m)}\right]_{k=1}^{K}$ from the predicted 3D lane. Different from the ground truth projected lane, the geometric transformation $\mathbf{G}$ is built by the estimated camera height h to transform 3D lanes onto the flat ground.

Geometry Constraint in Image Plane. Projecting lanes to this plane not only provides a supervision signal that is aligned with the input 2D image for free but also makes the 3D lanes and camera pose geometrically consistent with each other. More importantly, a small 3D jitter could cause a significant 2D shift. Such an amplification phenomenon undoubtedly facilitates the further optimization of 3D outputs. With the same optimal $\hat{l}$, the fitting loss in the image-plane is:

$$L_{img} = \sum_{m=1}^{M} \mathbb{1}(\hat{c}_m = 1)\,\alpha_2 \left|\hat{\mathbf{l}}_m - \mathbf{l}_{\hat{l}(m)}\right|. \tag{9}$$

In particular, $\hat{\mathbf{l}}_m = [\hat{u}_{km}, \hat{v}_{km}]_{k=1}^{K}$ is the m-th ground truth lane in the image plane. It is obtained by projecting the ground truth lane on the flat ground plane to the image plane via the homography transformation, where the transformation matrix constructed by the ground truth camera pose according to Eq. 2. Similarly, the $\hat{l}(m)$-th prediction $\mathbf{l}_{\hat{l}(m)} = \left[u_{k\hat{l}(m)}, v_{k\hat{l}(m)}\right]_{k=1}^{K}$ is calculated by the predicted 3D lane and homography transformation matrix $\mathbf{H}$ based on the estimated camera pose h and pitch angle ϕ.

Multi-task Joint Training Strategy. During training, we apply a multi-step training strategy to make optimization more stable. Specifically, we first train stage 1 and stage 2 separately, while stage 2 is fed with ground truth camera pose. Next, the whole architecture is jointly trained sequentially by feeding camera pose from stage 1 to stage 2. The IPM is a differentiable operation that can perform the transformation for either images or features. Thus, we could share parameters for two stages by applying IPM on features, enabling the part of the network before IPM to be reused for two stages to seek a better accuracy-cost trade-off. *Detailed studies can be found at the appendix.*

Scene	Method	CP	Height	Pitch	F-Score	AP	X error near	X error far	Z error near	Z error far
BS	3D-LaneNet	GT	-	-	86.4	89.3	0.068	0.477	0.015	**0.202**
	Gen-LaneNet	GT	-	-	88.1	90.1	**0.061**	0.496	**0.012**	0.214
	Stage 1 (ours)	\	0.021	0.121°	88.2	90.3	0.092	0.507	0.038	0.277
	CLGo (ours)	PD	0.021	0.121°	**91.9**	**94.2**	**0.061**	**0.361**	0.029	0.250
ROS	3D-LaneNet	GT	-	-	72.0	74.6	0.166	0.855	0.039	**0.521**
	Gen-LaneNet	GT	-	-	78.0	79.0	**0.139**	0.903	**0.030**	0.539
	Stage 1 (ours)	\	0.042	0.303°	77.1	78.9	0.193	0.919	0.077	0.679
	CLGo (ours)	PD	0.042	0.303°	**86.1**	**88.3**	0.147	**0.735**	0.071	0.609
SVV	3D-LaneNet	GT	-	-	72.5	74.9	0.115	0.601	0.032	**0.230**
	Gen-LaneNet	GT	-	-	85.3	87.2	**0.074**	0.538	**0.015**	0.232
	Stage 1 (ours)	\	0.026	0.155°	84.2	85.9	0.117	0.519	0.044	0.317
	CLGo (ours)	PD	0.026	0.155°	**87.3**	**89.2**	0.084	**0.464**	0.045	0.312

Table 1: Comparisons on 3D lane synthetic dataset testing set (%). The Height error goes in centimeters and X error and Z error are given in meters. Pitch error is given in angular ∘. CP, GT and PD are abbreviations of camera pose, ground truth and prediction. The best results are in bold.

Method	FPS	MACs	#Para	PP
3D-LaneNet	53	60.47	20.6	✓
Gen-LaneNet	60	9.85	3.4	✓
CLGo (ours)	**75**	**0.497**	**1.5**	×

Table 2: Comparisons of resource consumption. The number of MACs and parameters (#Para) are given in GHz and million. The PP means the requirement of post processing like outlier removal and non-maximum suppression.

Experiments

Datasets. We adopt the **ONLY** public 3D lane detection benchmark named 3D Lane Synthetic Dataset (Guo et al. 2020). The dataset consists of about 10,500 high-quality 1080×1920 images, containing abundant visual elements built by the unity game engine. The dataset exhibits highly diverse 3D worlds with realistic scenes across highways, urban and residential areas within the silicon valley in the United States under different weather conditions (morning, noon, evening). The camera intrinsic parameters are fixed, and the camera extrinsic parameters only differ in camera heights and camera pitch angles, which have the range of 1.4-1.8 meters and $0° - 10°$. The dataset is split originally into three different scenes: (1) Balanced Scenes (BS), (2) Rarely Observed Scenes (ROS), and (3) Scenes with Visual Variations (SVV). The SVV scene is used to conduct ablation studies since it covers illumination changes that affect camera pose estimation significantly.

Since there is no public realistic dataset for 3D lane evaluation, we provide a self-collected dataset named Forward-view Lane and Camera Pose (FLCP) to be the first one. It contains 1,287 images, each of which has corresponding 2D lane labels and calibrated camera poses. The test pipeline is to apply 3D detectors on FLCP images and project 3D detections into 2D results based on camera poses for evaluations.

Evaluation Metric. For 3D Lane Synthetic Dataset, we use the standard evaluation metric designed by Gen-LaneNet (Guo et al. 2020). Given the prediction set and ground truth set, an edit distance d_{edit} is defined to measure the lane-to-lane matching cost. For each prediction lane, it will be considered to be a true positive only if 75% of its covered y-positions have a point-wise distance less than the max length (1.5 meters). The percentages of matched ground-truth lanes and matched prediction lanes are reported as recall and precision. The Average Precision (AP) and maximum F-score are reported as a comprehensive evaluation and an evaluation of the best operation point. For the real-world FLCP Dataset, we use the CULane's F1 (Pan et al. 2018) for evaluation. In addition, we also report the FPS, MACs (Lyken17 2020) (one multiply-accumulate operation is approximated by two floating operations (FLOPs)), and the total number of network parameters.

Implementation Details. For a fair comparison with other methods, we also set the input resolutions of the image and the top-view image to 360×480 and 208×108, respectively. The flat ground plane has the range of $X \in [-10, 10] \times Y \in [1, 101]$ meters. The same Y-position sequence $\{3, 5, 10, 15, 20, 30, 40, 50, 65, 80, 100\}$ is used to re-sample lane points since the visual appearance in the distance gets sparser in top-view. The fixed number of predicted curves M and polynomial order R are set as 7 and 3. The batch size, training iterations and learning rate are 16, 450k, and 0.0001. α_1, α_2 and α_3 are set as 1, 5, 5.

Baselines. We treat the previous state-of-the-art **Gen-LaneNet** (Guo et al. 2020) and **3D-LaneNet** (Garnett et al. 2019) as competitors. Their results are dependent on perfect camera poses provided by the dataset. **CLGo** (**C**amera pose and 3D **L**ane with **G**ometry c**o**nstraints) is the final proposed method integrating Stage 1 and 2, which gets rid of ground truth camera poses during evaluation. Meanwhile, the **Stage 1** is also engaged because of no need perfect poses either.

Comparisons with State-of-the-Art Methods

In tab. 1, our CLGo achieves the highest F-Score and AP for all scenes without ground truth camera poses, and outperforms previous SOTA Gen-LaneNet for a large margin—

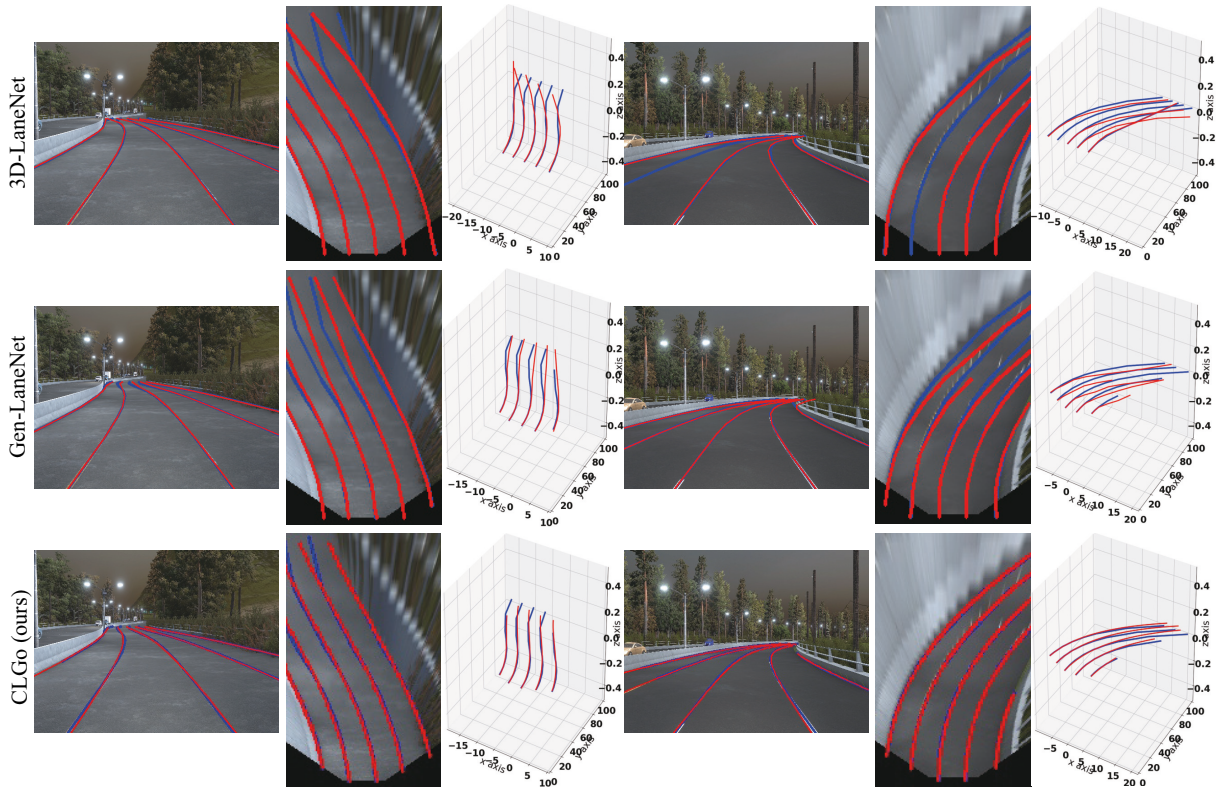

Figure 3: Qualitative comparisons with 3D-LaneNet (Garnett et al. 2019) and Gen-LaneNet (Guo et al. 2020) on the test set of balanced scenes. Red and blue lines indicate the estimation and ground truth lanes respectively. From right to left, we sequentially show the 3D fitting results and their projections on the flat ground plane and image plane.

3.8% and **4.1**% in balanced scene, **8.1**% and **9.3**% in rarely observed scene, **2.0**% and **2.0**% in scene with visual variations. Comparison results prove the effectiveness of the proposed methods for jointly learning camera pose and 3D lanes from a monocular camera. We notice that our local accuracy is comparable or a bit worse than SOTA, especially along Z-axis. This is because the fitting procedure requires unified coefficients to fit all local points simultaneously. Such a strategy may not be flexible enough to localize every local point accurately but is more reliable to capture the whole lane shape than the previous method (validated by F-Score and AP performance). Moreover, the poorer results of Stage 1 also prove the importance of camera poses in monocular 3D lane detection to transform the image into top-view.

Comparison of Efficiency. Tab. 2 demonstrates the comparison about the resource consumption. The CLGo works without ground truth camera poses and additional post-processing. Compared with Gen-LaneNet, our method consumes **2.26**$\times$ fewer parameters, **19.8**$\times$ less computation complexity and runs **1.25** $\times$ faster than it.

Qualitative Comparisons. The visualization of the lane detection results is given in Fig. 3. The left three columns demonstrate our method performs accurate fitting results, especially at the distant areas. As for the right three columns, our methods show complete and accurate lane estimations while the prediction of anchor-based methods was either incomplete (middle), or missed a lane entirely and wrongly clustered which causes lane crossing (top). Reliably perceiving lane structures completely is essential to keep driving safe, such as avoiding unwanted lane changes. We attribute improvements to (1) representing lanes as polynomials preserve smooth lane structures and embed global continuity to help fit the whole lane; (2) extracting non-local features is vital to learn lanes' long and thin structures for improving fitting performance, especially at remote regions.

Method	CP	F1	Prec.	Rec.
3D-LaneNet	GT	19.28	24.45	15.92
Gen-LaneNet	GT	30.87	40.72	24.86
Stage 1 (ours)	\	21.53	25.71	18.52
CLGo (ours)	PD	**33.82**	**40.80**	**28.88**

Table 3: Comparisons on the real-world FLCP dataset (%).

Evaluation on Real-world Images. Results in Tab. 3 show that our CLGo also has superior performance. The Stage 1 results are still poor, which validates the importance of using camera parameters to transform the viewpoint for realistic data. Fig. 4 shows qualitative comparisons. Facing unseen realistic scenes, our method predicates more reasonable, smooth, and continuous 3D lanes than others. Using the same camera poses, the projected 2D lanes of CLGo are

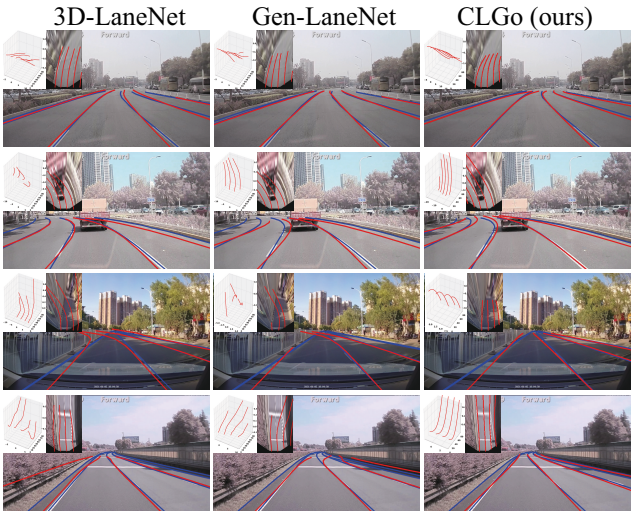

Figure 4: Qualitative comparisons on the self-collected real-world FLCP dataset. Red and blue lines indicate the estimation and ground truth lanes respectively.

Training	CP	F-Score	AP	Error
Jointly	GT	87.5	89.4	0.85
	PD	87.3	89.2	0.90
	dc	**-0.2**	**-0.2**	**+0.05**
Separately	GT	86.1	87.9	0.83
	PD	84.9	86.6	0.96
	dc	-1.2	-1.3	+0.13

Table 4: Comparison between using perfect and estimated camera poses on SVV (%). The lower the decreased value (dc), the better the performance.

also more complete and accurate. *Much more qualitative results on real scenes can be found in the appendix.*

Ablation Study

Comparison between Using Perfect and Estimated Camera Pose. To evaluate the performance of camera pose regression, we test the CLGo by using ground truth camera pose and analyze the changes. As Tab. 4 shows, when we jointly train two stages, the CLGo shows a very severe decline, which ensures the accuracy of camera pose regression. However, when we train stages separately, the performance shows a quite obvious degradation. Therefore, the multi-task joint training strategy benefits the performance a lot, which mainly comes from the improved tasks' compatibility. *More comparisons of previous methods among all scenes can be found in the appendix, which also shows that our jointly trained CLGo has the fewest decrease.*

Effect of Geometry Constraints. In this section, we gradually add geometry constraints to examine their contributions on two stages of CLGo. We attend to the pitch and ultimate 3D lane performances of CLGo, since the height error is extremely small in the order of magnitude (10^{-2} centimeters), which plays a small role. Tab. 5 shows the results of the proposed variations. The F-Score and AP drop 1.8% and 1.9% without geometry constraints comparing T2 and T4 (1.8% and 1.8% comparing T1 and T3), 1.0% and 1.1% without the auxiliary 3D lane branch comparing T3 and T4 (1.0% and 1.0% comparing T1 and T2). The reason for the above degradation is the deteriorating camera pose estimation, indicating that the proposed auxiliary branch and re-projection consistencies influence the most to the whole network.

	Stage 1 Loss				Stage 2 Loss		
Config	L_{cam}	L_{3D}	L_{top}	L_{img}	L_{3D}	L_{top}	L_{img}
T1	✓				✓		
T2	✓	✓			✓		
T3	✓		✓	✓	✓		
T4	✓	✓	✓	✓	✓		
T5	✓	✓	✓	✓	✓	✓	
T6	✓	✓	✓	✓	✓	✓	✓

Result	Pitch	Height	F-Score	AP	Error
T1	0.236°	0.041	84.5	86.3	1.10
T2	0.223°	0.033	85.5	87.3	1.00
T3	0.164°	0.027	86.3	88.1	0.96
T4	**0.155°**	0.026	**87.3**	**89.2**	**0.90**
T5	0.156°	**0.025**	87.1	89.0	0.91
T6	0.159°	0.028	85.3	87.0	1.03

Table 5: Evaluation of losses for two stages on SVV (%).

As for their effects on stage 2, the performance holds when adding constraint in the flat ground plane (T4 and T5), but with constrained points at the image plane (T6), the performance degrades significantly. We attribute the former to the top-view image already contains shape patterns about varied undulations, so such a constraint does not bring a significant effect. Meanwhile, we think the drop in the T6 is caused by viewpoint misalignment between the image plane constraint and top-view input. Many contextual clues such as skyline, trees, or buildings have been lost in the top-view, which decreases the network's ability to reconstruct lanes' constraints in the perspective-view image plane.

Conclusion

In this work, we present a full-vision-based 3D lane and camera pose estimation framework. The geometry constraints improve consistencies between 3D representations and 2D input by introducing mathematical geometry priors in model learning, as well as enhances compatibility between camera pose and 3D lane tasks, leading to significant improvements on both of them. The whole framework is validated with reliably estimated camera poses and outperforms state-of-the-art methods which are all evaluated using ground truth camera poses, while achieving the lightest model size, fewest computation costs, and the fastest FPS. It would be interesting to combine flexible representations for fitting lanes with complex topologies in future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61976170, 91648121, 62088102).

References

Bai, M.; Mattyus, G.; Homayounfar, N.; Wang, S.; Lakshmikanth, S. K.; and Urtasun, R. 2018. Deep Multi-Sensor Lane Detection. In *IROS*, 3102–3109.

Benmansour, N.; Labayrade, R.; Aubert, D.; and Glaser, S. 2008. Stereovision-based 3D lane detection system: a model driven approach. In *IEEE ITSC*, 182–188.

Brahmbhatt, S.; Gu, J.; Kim, K.; Hays, J.; and Kautz, J. 2018. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2616–2625.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *ECCV*, 213–229.

Chen, D.; Xu, D.; Li, H.; Sebe, N.; and Wang, X. 2018. Group Consistent Similarity Learning via Deep CRF for Person Re-Identification. In *CVPR*, 8649–8658.

Clark, R.; Wang, S.; Markham, A.; Trigoni, N.; and Wen, H. 2017. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *CVPR*, 2652–2660.

Coulombeau, P.; and Laurgeau, C. 2002. Vehicle yaw, pitch, roll and 3D lane shape recovery by vision. In *IEEE IVS*, 619–625 vol.2.

Efrat, N.; Bluvstein, M.; Garnett, N.; Levi, D.; Oron, S.; and Shlomo, B. E. 2020. Semi-Local 3D Lane Detection and Uncertainty Estimation. *CoRR*, abs/2003.05257.

Garnett, N.; Cohen, R.; Pe'er, T.; Lahav, R.; and Levi, D. 2019. 3D-LaneNet: End-to-End 3D Multiple Lane Detection. In *ICCV*, 2921–2930.

Guo, Y.; Chen, G.; Zhao, P.; Zhang, W.; Miao, J.; Wang, J.; and Choe, T. E. 2020. Gen-LaneNet: A Generalized and Scalable Approach for 3D Lane Detection. In *ECCV*, 666–681.

Homayounfar, N.; Ma, W.; Lakshmikanth, S. K.; and Urtasun, R. 2018. Hierarchical Recurrent Attention Networks for Structured Online Maps. In *CVPR*, 3417–3426.

Hou, Y.; Ma, Z.; Liu, C.; and Loy, C. C. 2019. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *ICCV*, 1013–1021.

Jin, Y.; Ren, X.; Chen, F.; and Zhang, W. 2021. Robust Monocular 3D Lane Detection With Dual Attention. In *IEEE ICIP*, 3348–3352.

Jung, S.; Choi, S.; Khan, M. A.; and Choo, J. 2020. Towards Lightweight Lane Detection by Optimizing Spatial Embedding. *CoRR*, abs/2008.08311.

Ko, Y.; Jun, J.; Ko, D.; and Jeon, M. 2020. Key Points Estimation and Point Instance Segmentation Approach for Lane Detection. *IEEE T-ITS*, 1–10.

Lee, M.; Lee, J.; Lee, D.; Kim, W. J.; Hwang, S.; and Lee, S. 2021. Robust Lane Detection via Expanded Self Attention. *CoRR*, abs/2102.07037.

Li, X.; Li, J.; Hu, X.; and Yang, J. 2020. Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit. *IEEE T-ITS*, 248–258.

Liu, R.; Yuan, Z.; Liu, T.; and Xiong, Z. 2021. End-to-end Lane Shape Prediction with Transformers. In *WACV*, 3693–3701.

Lyken17. 2020. MACs from THOP: PyTorch-OpCounter. https://github.com/Lyken17/pytorch-OpCounter.

Narote, S. P.; Bhujbal, P. N.; Narote, A. S.; and Dhane, D. M. 2018. A review of recent advances in lane detection and departure warning system. *Pattern Recognit.*, 216–234.

Neven, D.; Brabandere, B. D.; Georgoulis, S.; Proesmans, M.; and Gool, L. V. 2018. Towards End-to-End Lane Detection: an Instance Segmentation Approach. In *IEEE IVS*, 286–291.

Niu, J.; Lu, J.; Xu, M.; Lv, P.; and Zhao, X. 2016. Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting. *Pattern Recognit.*, 225–233.

Pan, X.; Shi, J.; Luo, P.; Wang, X.; and Tang, X. 2018. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI*, 7276–7283.

pandiii, y. 2021. Gen-LaneNet issues about why not using predicted camera poses. https://github.com/yuliangguo/Pytorch_Generalized_3D_Lane_Detection/issues/8.

Philion, J. 2019. FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network. In *CVPR*, 11582–11591.

Qin, Z.; Wang, H.; and Li, X. 2020. Ultra Fast Structure-aware Deep Lane Detection. In *ECCV*, 276–291.

Tabelini, L.; Berriel, R.; ao, T. M. P.; Badue, C.; Souza, A. F. D.; and Oliveira-Santos, T. 2021. Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. In *CVPR*, 294–302.

Torres, L. T.; Berriel, R. F.; Paixão, T. M.; Badue, C.; Souza, A. F. D.; and Oliveira-Santos, T. 2020. PolyLaneNet: Lane Estimation via Deep Polynomial Regression. In *ICPR*, 6150–6156.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Wang, Y.; Shen, D.; and Teoh, E. K. 2000. Lane detection using spline model. *Pattern Recognit. Lett.*, 677–689.

Xu, H.; Wang, S.; Cai, X.; Zhang, W.; Liang, X.; and Li, Z. 2020. CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending. In *ECCV*, 689–704.

Yoo, S.; Lee, H.; Myeong, H.; Yun, S.; Park, H.; Cho, J.; and Kim, D. H. 2020. End-to-End Lane Marker Detection via Row-wise Classification. In *CVPRW*, 4335–4343.

Zhang, J.; Xu, Y.; Ni, B.; and Duan, Z. 2018. Geometric Constrained Joint Lane Segmentation and Lane Boundary Detection. In *ECCV*, 502–518.

Zhao, Y.; Yuan, Z.; and Chen, B. 2020. Accurate Pedestrian Detection by Human Pose Regression. *IEEE T-IP*, 29: 1591–1605.

Zheng, T.; Fang, H.; Zhang, Y.; Tang, W.; Yang, Z.; Liu, H.; and Cai, D. 2020. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. In *AAAI*, 3547–3554.

Zhou, Y.; He, Y.; Zhu, H.; Wang, C.; Li, H.; and Jiang, Q. 2021. Monocular 3D Object Detection: An Extrinsic Parameter Free Approach. In *CVPR*, 7556–7566.