Gen-LaneNet: A Generalized and Scalable Approach for 3D Lane Detection

Yuliang Guo*, Guang Chen, Peitao Zhao, Weide Zhang,
Jinghao Miao, Jingao Wang, and Tae Eun Choe

Baidu Apollo, Sunnyvale CA 94089, USA

Abstract. We present a generalized and scalable method, called Gen-LaneNet, to detect 3D lanes from a single image. The method, inspired by the latest state-of-the-art 3D-LaneNet, is a unified framework solving image encoding, spatial transform of features and 3D lane prediction in a single network. However, we propose unique designs for Gen-LaneNet in two folds. First, we introduce a new geometry-guided lane anchor representation in a new coordinate frame and apply a specific geometric transformation to directly calculate real 3D lane points from the network output. We demonstrate that aligning the lane points with the underlying top-view features in the new coordinate frame is critical towards a generalized method in handling unfamiliar scenes. Second, we present a scalable two-stage framework that decouples the learning of image segmentation subnetwork and geometry encoding subnetwork. Compared to 3D-LaneNet, the proposed Gen-LaneNet drastically reduces the amount of 3D lane labels required to achieve a robust solution in real-world applications. Moreover, we release a new synthetic dataset and its construction strategy to encourage the development and evaluation of 3D lane detection methods. In experiments, we conduct extensive ablation study to substantiate the proposed Gen-LaneNet significantly outperforms 3D-LaneNet in average precision(AP) and F-measure.

Keywords: 3D lane detection, geometry-guided anchor, two-stage framework, monocular camera, unified network

1 Introduction

Over the past few years, autonomous driving has drawn numerous attention from both academic and industry. To drive safely, one of the fundamental problems is to perceive the lane structure accurately in real-time. Robust detection on current lane and nearby lanes is not only crucial for lateral vehicle control and accurate localization [14], but also a powerful tool to build and validate high definition map [8].

The majority of image-based lane detection methods treat lane detection as a 2D task [1, 4, 21]. A typical 2D lane detection pipeline consists of three components: A semantic segmentation component, which assigns each pixel in an image with a class label to indicate whether it belongs to a lane or not;

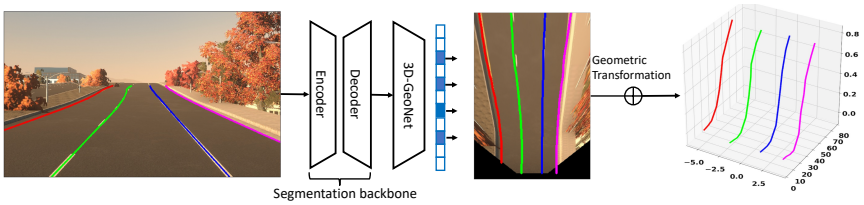


Fig. 1. Procedure of the Gen-LaneNet. A segmentation backbone(*image segmentation subnetwork*) first encodes an input image in deep features and decodes the features into a lane segmentation map. Given the segmentation as input, 3D-GeoNet(*geometry encoding subnetwork*) focuses on geometry encoding and predicts intermediate 3D lane points, specifically represented in top-view 2D coordinates and real heights. At last, the presented geometric transformation directly converts the network output to real-world 3D lane points.

a spatial transform component to project image segmentation output to a flat ground plane; and a third component to extract lanes which usually involves lane model fitting with strong assumption, *e.g.*, fitting quadratic curves. By assuming the world is flat, a 2D lane represented in the flat ground plane might be an acceptable approximation for a 3D lane in the ego-vehicle coordinate system. However, this assumption could lead to unexpected problems, as well studied in [6, 2]. For example, when an autonomous driving vehicle encounters a hilly road, an unexpected driving behavior is likely to occur since the 2D planar geometry provides incorrect perception of the 3D road.

To overcome the shortcomings associated with planar road assumption, the latest trend of methods [5, 19, 2, 6] has started to focus on perceiving complex 3D lane structures. Specifically, the latest state-of-the-art 3D-LaneNet [6] has introduced an end-to-end framework unifying image encoding, spatial transform between image view and top view, and 3D curve extraction in a single network. 3D-LaneNet shows promising results to detect 3D lanes from a monocular camera. However, representing lane anchors in an inappropriate space makes 3D-LaneNet not generalizable to unobserved scenes, while the end-to-end learned framework makes it highly affected by visual variations.

In this paper, we present **Gen-LaneNet**[1], a generalized and scalable method to detect 3D lanes from a single image. We introduce a new design of geometry-guided lane anchor representation in a new coordinate frame and apply a specific geometric transformation to directly calculate real 3D lane points from the network output. In principle our anchor design is an intuitive extension to the anchors of 3D-LaneNet, yet representing the lane anchor in an appropriate coordinate frame is critical for generalization. We demonstrate that aligning the anchor coordinates with the underlying top-view features in essence breaks down the global encoding of a whole scene to local patch level. Thus it makes the method more robust in handling unfamiliar scenes. Moreover, we present a scal-

[1] https://github.com/yuliangguo/Pytorch_Generalized_3D_Lane_Detection

able two-stage framework allowing the independent learning of image segmentation subnetwork and geometry encoding subnetwork, which drastically reduces the amount of 3D labels required for learning. Benefiting from more affordable 2D data, a two-stage framework outperforms end-to-end learnt framework when expensive 3D labels are rather limited to certain visual variations. Besides, we present a highly realistic synthetic dataset of images with rich visual variation, which would serve the development and evaluation of 3D lane detection. Finally in experiments, we conduct extensive ablation study to substantiate that the proposed Gen-LaneNet significantly outperforms state-of-the-art [6] in AP and F-measure, as high as 13% in some test sets.

2 Related Work

Various techniques have been proposed to tackle the lane detection problem. Driven by the effectiveness of Convolutional Neural Network(CNN), lots of recent progress can be observed in improving the 2D lane detection. Some prior methods focus on improving the accuracy of lane segmentation [7, 10, 13, 18, 25, 26, 17, 21, 9] while others try to improve segmentation and curve extraction in a unified network [16, 22]. More delicate network architectures are further developed to unify 2D lane detection and the following projection to planar road plane into an end-to-end learned network architecture [15, 12, 7, 20, 3]. However as discussed in Section 1, all these 2D lane detectors suffer from the specific planar world assumption. Indeed, even perfect 2D lanes are far from sufficient to imply accurate lane positions in 3D space.

As a better alternative, 3D lane detection methods assume no planar road and thus provide more reliable road perception. However, 3D lane detection is more challenging, because 3D information is generally unrecoverable from a single image. Consequently, existing methods are rather limited and usually based on multi-sensor or multi-view camera setups [5, 19, 2] rather than monocular camera. [2] takes advantage of both LiDAR and camera sensors to detect lanes in real world. But the high cost and high data sparsity of LiDAR limits its practical usage(*e.g.,* effective detection range is 48 meters in [2]). [5, 19] apply more affordable stereo cameras to perform the 3D lane detection, but they also suffer from low accuracy of 3D information in the distance.

The current state of the art, 3D-LaneNet [6], predicts 3D lanes from a single image. It has made a first attempt to solve 3D lane detection in a single network unifying image encoding, spatial transform of features and 3D curve extraction. It is realized in an end-to-end learning-based method with a network processing information in two pathways: The *image-view pathway* processes and preserves information from the image while the *top-view pathway* processes features in top-view to output the 3D lane estimations. Image-view pathway features are passed to the top-view pathway through four projective transformation layers which are conceptually built upon the spatial transform network [11]. Finally, top-view pathway features are fed into a *lane prediction head* to predict 3D lane points. Specifically, anchor representation of lanes has been developed to enable the

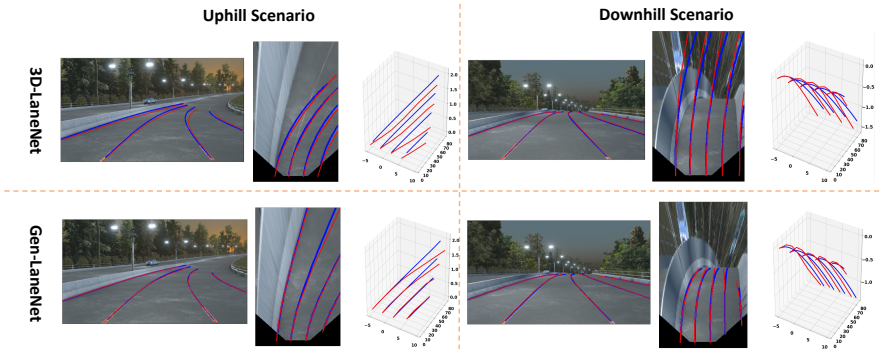


Fig. 2. Comparisons between 3D-LaneNet [6] and Gen-LaneNet in two typical scenes with ground height change. We have color-coded ground-truth lanes in blue and predicted lanes in red. Observed from **top-views** in each row, 3D-LaneNet represents anchor points in a coordinate frame not aligned with the underlying visual features(white lane marks). While the proposed Gen-LaneNet resolves this issue.

lane prediction head to estimate 3D lanes in the form of polylines. 3D-LaneNet shows promising results in recovering 3D structure of lanes in frequently observed scenes and common imaging conditions, however, its practicality is questionable due to two major drawbacks.

First, 3D-LaneNet uses an inappropriate coordinate frame to represent lane points in anchor representation, in which the ground truth of lane points is misaligned with visual features. This is most evident in the hilly road scenario, where the parallel lanes projected to the virtual top-view appear nonparallel, as observed in the top row of Fig. 2. However the ground-truth lanes (blue lines) are prepared in coordinates not aligned with the underlying visual features (white lane marks). Learning a model against such "corrupt" ground-truth could force the model sort to global encoding of the whole scene. This global encoding behavior in turn may cause the model not generalizable to a new scene partially different from an existing one of the training data.

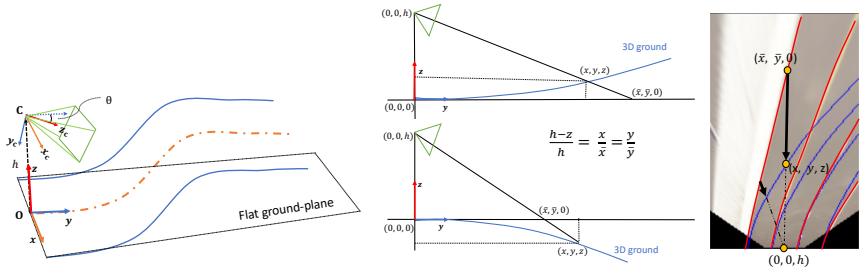
Second, the end-to-end learning-based framework indeed makes geometric encoding unavoidably affected by the change of image appearance, because it closely couples 3D geometry reasoning with image encoding. As a result, 3D-LaneNet might require exponentially increased amount of training data, in order to reason the same 3D geometry in the presence of occlusion by other traffic participants, varying lighting conditions in day and night, or different weather conditions. Unfortunately labeling 3D lanes is much more expensive than labeling 2D lanes in images. It often requires high-definition map built upon expensive multiple sensors (LiDAR, camera, etc), accurate localization and online calibration, and even the expensive sensor data manual alignment in 3D space to produce correct ground truth. This further prevents the 3D-LaneNet from being practical in real world.

3 Gen-LaneNet

Motivated by the success of 3D-LaneNet [6] and its drawbacks discussed in Section 2, we propose Gen-LaneNet, a generalized and scalable framework for 3D lane detection. Compared to 3D-LaneNet, Gen-LaneNet is still a unified framework that solves image encoding, spatial transform of features, and 3D curve extraction in a single network. But it involves major differences in two folds: a geometric extension to lane anchor design and a scalable two-stage network that decouples the learning of image encoding and 3D geometry reasoning.

3.1 Geometry in 3D Lane Detection

We begin by reviewing the geometry to establish the theory motivating our method. In a common vehicle camera setup as illustrated in Fig. 3(a), 3D lanes are represented in the ego-vehicle coordinate frame defined by $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ axes and origin $\boldsymbol{O}$. Specifically $\boldsymbol{O}$ defines the perpendicular projection of camera center on the road. Following a simple setup, only camera height h and pitch angle θ are considered to represent camera pose which leads to camera coordinate frame defined by $\boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{z_c}$ axes and origin $\boldsymbol{C}$. A *virtual top-view* can be generated by first projecting a 3D scene to the image plane through a projective transformation and then projecting the captured image to the flat road-plane via a planer homography. Because camera parameters are involved, points in the virtual top-view in principle have different x, y values compared to their corresponding 3D points in the ego-vehicle system. In this paper, we formally considers the virtual top-view as a unique coordinate frame defined by axes $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{z}$ and original $\mathbf{O}$. The geometric transformation between virtual top-view coordinate frame and ego-vehicle coordinate frame is derived next.



(a) (b) (c)

Fig. 3. Geometry in 3D lane detection. (a) Camera setup. **(b)** The co-linear relationship between a 3D lane point (x, y, z), its projection on the virtual top-view $(\bar{x}, \bar{y}, 0)$ and camera center $(0, 0, h)$ holds, no matter $z > 0$ (top) or $z < 0$ (bottom). **(c)** In the virtual top-view, estimating lane height z is conceptually equivalent to estimating the vector field(black arrows) moving top-view lane points (red curves) to their destination positions such that they can form parallel curves (blue curves).

For a projective camera, a 3D point (x, y, z), its projection on the image plane, and the camera optical center $(0, 0, h)$ should lie on a single ray. Similarly, if a point $(\bar{x}, \bar{y}, 0)$ from the virtual top-view is projected to the same image pixel, it must be on the same ray. Accordingly, camera center $(0, 0, h)$, a 3D point (x, y, z) and its corresponding virtual top-view point $(\bar{x}, \bar{y}, 0)$ appear to be co-linear, as shown in Fig. 3 (b) and (c). Formally, the relationship between these three points can be written as:

$$\frac{h-z}{h} = \frac{x}{\bar{x}} = \frac{y}{\bar{y}}. \tag{1}$$

Specifically, as illustrated in Fig. 3 (b), this relationship holds no matter z is positive or negative. Thus we derive the geometric transformation from virtual top-view coordinate frame to 3D ego-vehicle coordinate frame as:

$$\begin{aligned} x &= \bar{x} \cdot (1 - \frac{z}{h}) \\ y &= \bar{y} \cdot (1 - \frac{z}{h}), \end{aligned} \tag{2}$$

It is worth mentioning that the obtained transformation describes a general relationship without assuming zero yaw and roll angles in camera orientation.

3.2 Geometry-Guided Anchor Representation

Following the presented geometry, we solve 3D lane detection in two steps: A network is first applied to encode the image, transform the features to the virtual top-view, and predict lane points represented in virtual top-view; afterwards the presented geometric transformation is adopted to calculate 3D lane points in ego-vehicle coordinate frame, as shown in Fig. 5. Equation 2 in principle guarantees the feasibility of this approach because the transformation is shown to be independent from camera orientations. This is an important fact to ensures the approach not affected by the inaccurate camera pose estimation.

Anchor representation is the core of a network realization unifying boundary detection and contour grouping in a structured scene because it effectively constrains the search space to a tractable level. Similar to 3D-LaneNet [6], we develop an anchor representation such that a network can directly predict 3D lanes in the form of polylines. Formally, as shown in Fig. 4, lane anchors are defined as N equally spaced vertical lines in x-positions $\{X_A^i\}_{i=1}^N$. Given a set of pre-defined fixed y-positions $\{y_i\}_{j=1}^K$, each anchor X_A^i defines a 3D lane line in $3{\cdot}K$ attributes $\{(\bar{x}_j^i, z_j^i, v_j^i)\}_{j=1}^K$ or equivalently in three vectors as $(\mathbf{x}^i, \mathbf{z}^i, \mathbf{v}^i)$, where the values $\bar{x}_j^i$ are horizontal offsets relative to the anchor position and the attribute v_j^i indicates the visibility of every lane point. Denoting lane center-line type with c and lane-line type with l, each anchor can be written as $X_A^i = \{(\mathbf{x}_t^i, \mathbf{z}_t^i, \mathbf{v}_t^i, p_t^i)\}_{t\in\{c,l\}}$, where p_t^t indicates the existence probability of a lane. Based on this anchor representation, our network outputs 3D lane lines in the virtual top-view. The derived transformation is applied afterwards to calculate their corresponding 3D lane

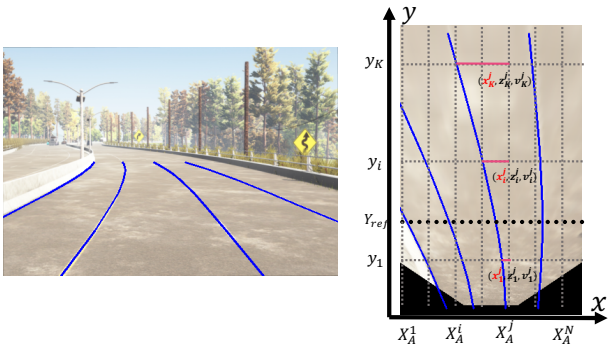


Fig. 4. Anchor representation. Lane anchors are defined as N equally spaced vertical lines in x-positions $\{X_A^i\}_{i=1}^N$. Given a set of pre-defined fixed y-positions $\{y_i\}_{j=1}^K$, a 3D lane can be represented with an anchor X_A^i composed of $3 \cdot K$ attributes $\{(\bar{x}_j^i, z_j^i, v_j^i)\}_{j=1}^K$. Specifically, $\bar{x}_j^i$ indicates x-position in the virtual top-view. A ground-truth lane is associated with its closest anchor based on x-value at Y_{ref}.

points. Given predicted visibility probability per lane point, only those visible lane points will be kept in the final output.

Our anchor representation involves two major changes compared to 3D-LaneNet. First, lane point positions are represented in a different coordinate frame, the virtual top-view. This change guarantees the target lane position to align with projected image features, as shown in the bottom row of Fig. 2. Compared to the global encoding of the whole scene as in 3D-LaneNet, establishing the correlation at local patch-level is more robust to novel or unobserved scenes. Even a new scene's overall structure has not been observed from training, those local patches more likely have been. Second, additional attributes are introduced to the representation to indicate the visibility of each anchor point. As a result, our method is more stable in handling partially visible lanes starting or ending in halfway, as observed in Fig. 2.

3.3 Two-Stage Framework with Decoupled Image Encoding and Geometry Reasoning

Instead of adopting an end-to-end learned network, we propose a two-stage framework which decouples the learning of image encoding and 3D geometry reasoning. Basically, the two-stage framework relieves the dependence of 3D geometry on image appearance via introducing an intermediate representation in the form of 2D lane segmentation. As shown in Fig. 5, the first subnetwork focuses on lane segmentation in image domain; the second predicts 3D lane structure from the segmentation outputs of the first subnetwork. The two-stage framework is well motivated by an important fact that the encoding of 3D geometry is rather independent from image features. As observed from Fig. 3 (b), ground height z is mostly correlated to the displacement vector from the position

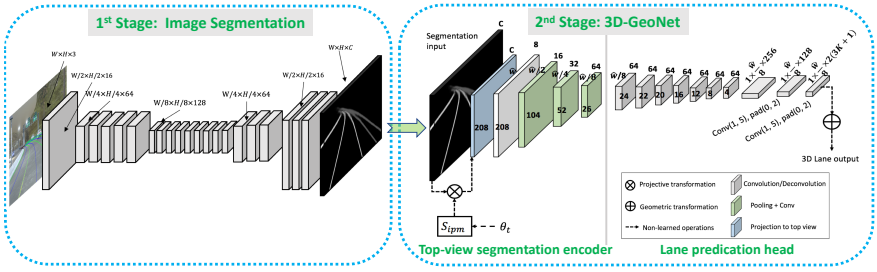


Fig. 5. The two-stage network. An input image is first fed into the image segmentation subnetwork to generate a lane segmentation map with the same resolution. The intermediate segmentation map then goes through 3D-GeoNet, which is composed of the top-view segmentation encoder and the lane prediction head. The output 3D lanes are represented in the virtual top-view. At last, the presented geometric transformation is applied to calculate 3D lane points in the ego-vehicle system.

$(\bar{x}, \bar{y})$ to position (x, y). Therefore, estimating the ground heights is conceptually equivalent to estimating a vector field such that all the points corresponding to lanes in the top-view are moved to positions overall in parallelism. It can be anticipated that the geometric information carried by 2D lane segmentation suffices for the 3D lane prediction.

There are a bunch of off-the-shelves candidates [24, 23, 21, 9] to perform 2D lane segmentation in image, any of which could be effortlessly integrated to the first stage of our framework. Although contemporary methods achieve higher performance, we choose ERFNet [24] for its simplicity hence to emphasize the raw power of the two-stage framework. For 3D lane prediction, we introduce **3D-GeoNet**, as shown in Fig. 5, to estimate 3D lanes from image segmentation. The segmentation map is first projected to the top-view and encoded into a top-view feature map through the *top-view segmentation encoder*. Then the *lane prediction head* recovers 3D lane attributes based on the proposed anchor representation. 3D Lane points produced by *lane prediction head* are represented in top-view positions, while 3D lane points in ego-vehicle coordinate frame are calculated afterwards through the introduced geometric transformation.

Decoupling the learning of image encoding and geometry reasoning makes the two-stage framework more cost-effective and scalable. As discussed in Section 2, an end-to-end learned framework like [6] is closely keen to image appearance. Consequently, it depends on huge amount of very expensive real-world 3D data for the leaning. On contrary, the two-stage pipeline drastically reduces the cost as it no longer requires to collect redundant real 3D lane labels in the same area under different weathers, day times, and occlusion cases. Moreover, the two-stage framework could leverage on more sufficient 2D real data, *e.g.,* [4, 1, 21], to train a more reliable 2D lane segmentation subnetwork. With extremely robust segmentation as input, 3D lane prediction would in turn perform better. In an optimal situation, the two-stage framework could train the image segmentation subnetwork from 2D real data and train the 3D geometry subnetwork with only

synthetic 3D data. We postpone the optimal solution as future work because domain transfer technique is required to resolve the domain gap between perfect synthetic segmentation ground truth and segmentation output from the first subnetwork.

3.4 Training

Given an image and its corresponding ground-truth 3D lanes, the training proceeds as follows. Each ground-truth lane curve is projected to the virtual top-view, and is associated with the closest anchor at Y_{ref}. The ground-truth anchor attributes are calculated based on the ground-truth values at the pre-defined y-positions $\{y_i\}_{j=1}^{K}$. Given pairs of predicted anchor X_A^i and corresponding ground-truth $\hat{X}_A^i = \{(\hat{\mathbf{x}}_t^i, \hat{\mathbf{z}}_t^i, \hat{\mathbf{v}}_t^i, \hat{p}_t^i)\}_{t\in\{c,l\}}$, the loss function can be written as:

$$\begin{aligned}\ell = &- \sum_{t\in\{c,l\}} \sum_{i=1}^{N} (\hat{p}_t^i \log p_t^i + (1-\hat{p}_t^i)\log(1-p_t^i)) \\ &+ \sum_{t\in\{c,l\}} \sum_{i=1}^{N} \hat{p}_t^i \cdot (\|\hat{\mathbf{v}}_t^i \cdot (\mathbf{x}_t^i - \hat{\mathbf{x}}_t^i)\|_1 + \|\hat{\mathbf{v}}_t^i \cdot (\mathbf{z}_t^i - \hat{\mathbf{z}}_t^i)\|_1) \\ &+ \sum_{t\in\{c,l\}} \sum_{i=1}^{N} \hat{p}_t^i \cdot \|\mathbf{v}_t^i - \hat{\mathbf{v}}_t^i\|_1 \end{aligned} \tag{3}$$

There are three changes compared to the loss function introduced in 3D-LaneNet [6]. First, both $\mathbf{x}_t^i$ and $\hat{\mathbf{x}}_t^i$ are represented in virtual top-view coordinate frame rather than the ego-vehicle coordinate frame. Second, additional cost terms are added to measure the difference between predicted visibility vector and ground-truth visibility vector. Third, cost terms measuring $\bar{x}$ and z distances are multiplied by its corresponding visibility probability v such that those invisible points do not contribute to the loss.

4 Synthetic Dataset and Construction Strategy

Due to lack of 3D lane detection benchmark, we construct a synthetic dataset to develop and validate 3D lane detection methods. Our dataset[2] simulates abundant visual elements and specifically focuses on evaluating a method's generalization capability to rarely observed scenarios. We use Unity game engine to build highly diverse 3D worlds with realistic background elements and render images with diversified scene structure and visual appearance.

The synthetic dataset is rendered from three world maps with diverse terrain information. All the maps are based on real regions within the silicon valley in the United States. Lane lines and center lines involve adequate ground height variation and turnings, as shown in Fig. 6. Images are sparsely rendered at

[2] https://github.com/yuliangguo/3D_Lane_Synthetic_Dataset

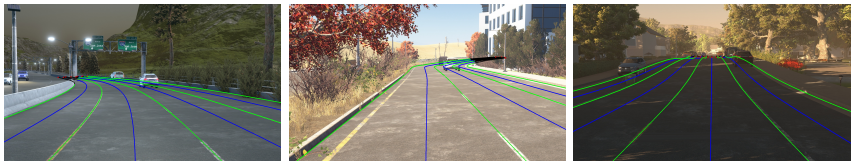


Fig. 6. Examples of synthetic data. Images in the dataset are rendered from different world maps with diverse day-times respectively. In each image, lane lines and center lines are drawn in green and blue separately. Those black-colored segments of lanes in the distance are discarded in a post-process, as background-occluded segments are generally not desired from a lane detection method.

different locations and different day-times(morning, noon, evening), under two levels of lane-marker degradation, random camera-height within $1.4 \sim 1.8m$ and random pitch angles within $0° \sim 10°$. We used fixed intrinsic parameters during data rendering and placed a decent amount of agent vehicles driving in the simulation environment, such that the rendered images include realistic occlusions of lanes. In summary, a total of 6000 samples from virtual highway map, 1500 samples from urban map, and 3000 samples from residential area, along with corresponding depth map, semantic segmentation map, and 3D lane lines information are provided. 3D lane labels are truncated at 200 meters distance to the camera, and at the border of the rendered image.

So far, essential information about occlusion is still missing for developing reliable 3D lane detectors. In general, a lane detector is expected to recover the foreground-occluded portion but discard the background-occluded portion of lanes, which in turn requires accurate labeling of the occlusion type per lane point. In our dataset, we use ground-truth depth maps and semantic segmentation maps to deduce the occlusion type of lane points. First, a lane point is considered occluded when its y position is deviated from the value at the corresponding pixel in the depth map. Second, its occlusion type is further determined based on semantic segmentation map. The final dataset keeps the portion of lanes occluded by foreground but discard the portion occluded by background, as the black segments in the distance shown in Fig. 6.

5 Experiments

In the section, we first describe the experimental setups, including dataset splits, baselines, algorithm implementation details, and evaluation metrics. Then we conduct experiments to demonstrate our contributions in ablation. Finally, we design and conduct experiments to substantiate the advantages of our method, compared with prior state of the art [6].

5.1 Experimental Setup

Dataset Setup: In order to evaluate algorithms from different perspectives, we design three different rules to split the synthetic dataset:

(1) *Balanced scenes:* the training and testing set follow a standard five-fold split of the whole dataset, to benchmark algorithms with massive, unbiased data.

(2) *Rarely observed scenes:* This dataset split contains the same training data as *balanced scenes*, but uses only a subset of the testing data, captured from the complex urban map. This dataset split is designed to examine a method's capability of generalization to the test data rarely observed from training. Because the testing images are sparsely rendered at different locations involving drastic elevation change and sharp turnings, the scenes in testing data are rarely observed from the training data.

(3) *Scenes with visual variations:* This split of dataset evaluates methods under the change of illumination, assuming more affordable 2D data compared to expensive 3D data is available to cover the illumination change for the same region. Specifically, the same training set as *balanced scenes* is used to train image segmentation subnetwork in the first stage of our Gen-LaneNet. However 3D examples from a certain day time, namely before dawn, are excluded from the training of 3D geometry subnetwork of our method(3D-GeoNet) and 3D-LaneNet [6]. In testing, on contrary, only examples corresponding to the excluded day time are used.

Baselines and Parameters: Gen-LaneNet is compared to two other methods: Prior state-of-the-art 3D-LaneNet [6] is considered as a major baseline; To honestly study the upper bound of our two-stage framework, we treats 3D-GeoNet subnetwork as a stand-alone method which is fed with ground-truth 2D lane segmentation. To conduct fair comparison, all the methods resize the original image into size 360×480 and use the same spatial resolution 208×108 for the first top-view layer to represent a flat-ground region with the range of $[-10, 10] \times [1, 101]$ meters along $\boldsymbol{x}$ and $\boldsymbol{y}$ axes respectively. For the anchor representation, we use y-positions $\{3, 5, 10, 15, 20, 30, 40, 50, 65, 80, 100\}$, where the intervals are gradually increasing due to the fact that visual information in the distance gets sparser in top-view. In label preparation, we set $Y_{ref} = 5$ to associate each lane label with its closest anchor. In training, all the networks are randomly initialized with normal distribution and trained from scratch with Adam optimization and with an initial learning rate $5 \cdot 10^{-4}$. We set batch size 8 and complete training in 30 epochs. For training ERFNet, we follows the same procedure described in [24], but with modified input image size and output segmentation maps sizes. To rule out the error caused by inaccurate camera parameters, we conduct all the experiments given perfect camera intrinsic and extrinsic parameters provided by the synthetic dataset.

Evaluation Metrics: We formulate the evaluation of 3D lane detection as a bipartite matching problem between predicted lanes and ground-truth lanes. The global best matching is sought via minimum-cost flow. Our evaluation method is so far the most strict compared to one-to-many matching in [1] or greedy search bipartite matching in [6].

To handle partial matching properly, we define a new pairwise cost between lanes in euclidean distance. Specifically, lanes are represented in $X^j = \{x_i^j, z_i^j, v_i^j\}_{i=1}^n$ at n pre-determined y-positions, where v^i indicates whether the

y-position is covered by a given lane. Denser y-positions compared to the anchor points are used here, which are equally placed from 0 to 100 meters with 2 meter interval. Formally, the lane-to-lane cost between X^j and X^k is calculated as the square root of the squared sum of point-wise distances over all y-positions, written as $cost_{jk} = \sqrt{\sum_i^n d_i^{jk}}$, where

$$d_i^{jk} = \begin{cases} (x_i^j - x_i^k)^2 + (z_i^j - z_i^k)^2, & \text{if } v_i^j = 1 \text{ and } v_i^k = 1 \\ 0, & \text{if } v_i^j = 0 \text{ and } v_i^k = 0 \\ d_{max}, & \text{otherwise.} \end{cases}$$

Specifically, point-wise euclidean distance is calculated when a y-position is covered by both lanes. When a y-position is only covered by one lane, the point-wise distance is assigned to a max-allowed distance $d_{max} = 1.5m$. While a y-position is not covered by any of the lanes, the point-wise distance is set to zero. Following such metric, a pair of lanes covering different ranges of y-positions can still be matched, but at an additional cost proportional to the number of *edited* points. This defined cost is inspired by the concept of *edit distance* in string matching. After enumerating all pairwise costs between two sets, we adopt the solver included in Google OR-tools to solve the minimum-cost flow problem. Per lane from each set, we consider it matched when 75% of its covered y-positions have point-wise distance less than the max-allowed distance (1.5 meters).

At last, the percentage of matched ground-truth lanes is reported as recall and the percentage of matched predicted lanes is reported as precision. We report both the Average Precision(AP) as a comprehensive evaluation and the maximum F-score as an evaluation of the best operation point in application.

5.2 Anchor Effect

		balanced scenes			**rarely observed scenes**			**scenes with visual variations**		
		w/o	w/	gain	w/o	w/	gain	w/o	w/	gain
3D-LaneNet	**F-score**	86.4	90.0	+3.6	72.0	80.9	+8.9	72.5	82.7	+10.5
	AP	89.3	92.0	+2.7	74.6	82.0	+7.4	74.9	84.8	+9.9
3D-GeoNet	**F-score**	88.5	91.8	+3.3	75.4	84.7	+9.3	83.8	90.2	+6.4
	AP	91.3	93.8	+2.5	79.0	86.6	+7.6	86.3	92.3	+6.0
Gen-LaneNet	**F-score**	85.1	88.1	+3.0	70.0	78.0	+8.0	80.9	85.3	+4.4
	AP	87.6	90.1	+2.5	73.0	79.0	+6.0	83.8	87.2	+3.4

Table 1. The comparison in anchor representations. "w/o" represents the integration with anchor design in [6], while "w" represents the integration with our anchor design. On three dataset splits, we show the performance gain by integrating our anchor design.

We first demonstrate the superiority of the presented geometry-guided anchor representation compared to [6]. For each candidate method, we keep the

architecture exact the same, except the anchor representation integrated. As reported in Table 1, all the three methods, no matter end-to-end 3D-LaneNet [6], "theoretical existing" 3D-GeoNet, or our two-stage Gen-LaneNet, benefit significantly from the new anchor design. Both AP and F-score achieve 3% to 10% improvements, across all splits of dataset.

5.3 The Upper Bound of the Two-Stage Framework

Experiments are designed to substantiate that a two-stage method potentially gains higher accuracy when more robust image segmentation is integrated, and meanwhile to localize the upper bound of Gen-LaneNet when perfect image segmentation subnetwork is provided. As shown in Table 2, 3D-GeoNet consistently outperforms Gen-LaneNet and 3D-LaneNet across all three experimental setups. We notice that on *balanced scenes*, the improvement over Gen-LaneNet is pretty obvious, around 3% better, while on *rarely observed scenes* and *scenes with visual variations*, the improvement is significant from 5% to 7%. This observation is rather encouraging because the 3D geometry from hard cases(*e.g.,*new scenes or images with dramatic visual variations) can still be reasoned well from the abstract ground-truth segmentation or from the output of image segmentation subnetwork. Besides, Table 2 also shows promising upper bound of our method, as the 3D-GeoNet outperforms 3D-LaneNet [6] by a large margin, from 5% to 18% in F-score and AP.

		balanced scenes	**rarely observed scenes**	**scenes with visual variations**
3D-LaneNet	**F-score**	86.4	72.0	72.5
	AP	89.3	74.6	74.9
3D-GeoNet	**F-score**	**91.8**	**84.7**	**90.2**
	AP	**93.8**	**86.6**	**92.3**
Gen-LaneNet	**F-score**	**88.1**	**78.0**	**85.3**
	AP	**90.1**	**79.0**	**87.2**

Table 2. The upper bound of the two-stage framework, represented by 3D-GeoNet, outperforms the 3D-LaneNet [6] significantly on all the three dataset splits.

5.4 Whole System Evaluation

We conclude our experiments with the whole system comparison between our two-stage Gen-LaneNet with prior state-of-the-art 3D-LaneNet [6]. The apple-to-apple comparisons have been taken on all the three splits of dataset, as shown in Table 3. On the *balanced scenes* the 3D-LaneNet works well, but our Gen-LaneNet still achieves 0.8% AP and 1.7% F-score improvement. Considering this data split is well balanced between training and testing data and covers various scenes, it means the proposed Gen-LaneNet have better generalization on various

scenes; On the *rarely observed scenes*, both AP and F-score are improved 6% and 4.4% respectively by our method, demonstrating the superior robustness of our method when it meets uncommon test scenes; Finally on the *scenes with visual variations*, our method significantly surpasses the 3D-LaneNet by around 13% in F-score and AP, which shows that our two-stage algorithm successfully benefits from the decoupled learning of the image encoding and 3D geometry reasoning. For any specific scene, we could annotate more cost-effective 2D lanes in image, to learn a general segmentation subnetwork while label a limited number of expensive 3D lanes to learn the 3D lane geometry. This makes our method a more scalable solution in real-world application. Qualitative comparisons are presented in the supplemental material.

Besides F-score and AP, errors (euclidean distance) in meters over those matched lanes are respectively reported for **near range**(0-40m) and **far range**(40-100m). This is a complementary evaluation focusing on the quality of the detected portion. As observed, Gen-LaneNet maintains the error lower or on par with 3D-LaneNet, even more matched lanes are involved[3].

Running Time Analysis: The average running speed of Gen-LaneNet is 60 FPS on a single NVIDIA RTX 2080 GPU, compared to 53 FPS of 3D-LaneNet.

Dataset Splits	Method	F-Score	AP	x error near (m)	x error far (m)	z error near (m)	z error far (m)
balanced scenes	3D-LaneNet	86.4	89.3	0.068	0.477	0.015	0.202
	Gen-LaneNet	88.1	90.1	0.061	0.496	0.012	0.214
rarely observed scenes	3D-LaneNet	72.0	74.6	0.166	0.855	0.039	0.521
	Gen-LaneNet	78.0	79.0	0.139	0.903	0.030	0.539
scenes with visual variations	3D-LaneNet	72.5	74.9	0.115	0.601	0.032	0.230
	Gen-LaneNet	85.3	87.2	0.074	0.538	0.015	0.232

Table 3. Whole system comparison between 3D-LaneNet [6] and Gen-LaneNet.

6 Conclusion

We present a generalized and scalable 3D lane detection method, Gen-LaneNet. A geometry-guided anchor representation has been introduced together with a two-stage framework decoupling the learning of image segmentation and 3D lane prediction. Moreover, we present a new strategy to construct synthetic dataset for 3D lane detection. We experimentally substantiate that our method surpasses 3D-LaneNet significantly in both AP and in F-score from various perspectives.

Acknowledgements
This work was supported by Apollo autonomous driving solution, Baidu USA.

[3] A method with higher F-score calculates the errors from more matched lane points.

References

1. Tusimple. In: https://github.com/TuSimple/tusimple-benchmark (2018)
2. Bai, M., Máttyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S.K., Urtasun, R.: Deep multi-sensor lane detection. CoRR (2019)
3. Brabandere, B.D., Gansbeke, W.V., Neven, D., Proesmans, M., Gool, L.V.: End-to-end lane detection through differentiable least-squares fitting. CoRR **abs/1902.00293** (2019)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. pp. 3213–3223 (06 2016)
5. Coulombeau, P., Laurgeau, C.: Vehicle yaw, pitch, roll and 3d lane shape recovery by vision. In: Intelligent Vehicle Symposium, 2002. IEEE (2002)
6. Garnett, N., Cohen, R., Pe'er, T., Lahav, R., Levi, D.: 3d-lanenet: end-to-end 3d multiple lane detection. In: IEEE International Conference on Computer Vision, ICCV (2019)
7. He, B., Ai, R., Yan, Y., Lang, X.: Accurate and robust lane detection based on dual-view convolutional neutral network. In: Intelligent Vehicles Symposium (2016)
8. Homayounfar, N., Ma, W., Lakshmikanth, S.K., Urtasun, R.: Hierarchical recurrent attention networks for structured online maps. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 3417–3426 (2018)
9. Hou, Y., Ma, Z., Liu, C., Loy, C.C.: Learning lightweight lane detection cnns by self attention distillation. In: IEEE International Conference on Computer Vision, ICCV (2019)
10. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F.A., Coates, A., Ng, A.Y.: An empirical evaluation of deep learning on highway driving. CoRR (2015)
11. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. pp. 2017–2025 (2015)
12. Kheyrollahi, A., Breckon, T.P.: Automatic real-time road marking recognition using a feature driven approach. Mach. Vision Appl. pp. 123–133 (2012)
13. Kim, J., Lee, M.: Robust lane detection based on convolutional neural network and random sample consensus. ICONIP (2014)
14. Kogan, V., Shimshoni, I., Levi, D.: Lane-level positioning with sparse visual cues. In: 2016 IEEE Intelligent Vehicles Symposium, IV 2016, Gotenburg, Sweden, June 19-22, 2016. pp. 889–895 (2016)
15. Lee, S., Kim, J., Yoon, J.S., Shin, S., Bailo, O., Kim, N., Lee, T., Hong, H.S., Han, S., Kweon, I.S.: Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 1965–1973 (2017)
16. Li, J., Mei, X., Prokhorov, D.: Deep neural network for structural prediction and lane detection in traffic scene. IEEE transactions on neural networks and learning systems (2016)
17. Meyer, A., Salscheider, N., Orzechowski, P., Stiller, C.: Deep semantic lane segmentation for mapless driving. In: IROS (2018). https://doi.org/10.1109/IROS.2018.8594450

18. Mriut, F., Foalu, C., Petrisor, D.: Lane mark detection using hough transform. In: 2012 International Conference and Exposition on Electrical and Power Engineering. pp. 871–875 (2012)
19. Nedevschi, S., Schmidt, R., Graf, T., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C.: 3d lane detection system based on stereovision. In: Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749). pp. 161–166 (2004)
20. Neven, D., Brabandere, B.D., Georgoulis, S., Proesmans, M., Gool, L.V.: Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018. pp. 286–291 (2018)
21. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial CNN for traffic scene understanding. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 7276–7283 (2018)
22. Philion, J.: Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 11582–11591 (2019)
23. Poudel, R.P.K., Bonde, U., Liwicki, S., Zach, C.: Contextnet: Exploring context and detail for semantic segmentation in real-time. BMVC (2018)
24. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Trans. Intelligent Transportation Systems (2018)
25. Yuan, C., Chen, H., Liu, J., Zhu, D., Xu, Y.: Robust lane detection for complicated road environment based on normal map. IEEE Access (2018)
26. Zhang, W., Mahale, T.: End to end video segmentation for driving : Lane detection for autonomous car. CoRR (2018)