Data Driven 3D-Lane Detection Using Parallelism Loss Function

Mohammed Hassoubah [1,*] and Ganesh Sistu [2]

[1] Department of Comfort Driving Assistance, Valeo, Cairo, Egypt
[2] Valeo Vision Systems, Tuam, Ireland
Email: ganesh.sistu@valeo.com (G.S.); mohammed.hassoubah@valeo.com (M.H.)
*Corresponding author

***Abstract*—Accurate lane position prediction is crucial in autonomous driving for safe vehicle maneuvering. Monocular cameras, aided by AI advancements, have proven to be effective in this task. However, 2D image space predictions overlook lane height, causing poor results in uphill or downhill scenarios that affect action judgments, such as in the planning and control module. Previous 3D-lane detection approaches relied solely on applying Inverse Perspective Mapping (IPM) on the encoded camera feature map, which may not be ordered according to the perspective principle leading to sub-optimal prediction results. To address these issues, we present the LS-3DLane network, inspired by the Lift-Splat-Shoot architecture, which predicts lane position in 3D space using a data-driven approach. The network also employs the Parallelism loss, using prior knowledge of lane geometry, to improve performance. Such loss can be used for training any 3D lane position prediction network and would boost the performance. Our results show that LS-3DLane outperforms previous approaches like Gen-LaneNet and 3D-LaneNet, with F-score improvements reaching 5.5% and 10%, respectively, in certain cases. LS-3DLane performs similarly in X/Z error metrics. Parallelism loss was shown to boost the F-Score KPI when applied to any of the models under test (LS-3DLane, GenLaneNet, and 3D-LaneNet) by up to 2.8% in certain cases and has a positive impact on nearly all the other KPIs.**

***Keywords*—monocular camera, 3D-Lanes detection, lift-splat-shot, anchor, geometric structure**

I. Introduction

One of the most widely used cases of AI algorithms in recent years is autonomous driving. For L2 and beyond, contemporary Advanced Driver Assistance Systems (ADAS) include functionalities like Lane Keep Assist (LKA) and Lane Departure Warning (LDW). The fundamental requirement for perception is reliable and universal lane line detection [1]. Lane identification algorithms in the 2D image space have produced excellent results because of advancements in deep learning [2, 3]. The job is formulated as 2D segmentation given a front-facing camera image (perspective) as an input [4, 5]. However, the major issue with this approach arises when the detected lanes in the image are projected to world coordinates. A 2D lane represented in the flat ground plane might be a good approximation for a 3D lane in the ego-vehicle coordinate system if the world is assumed to be flat. However, as Garnett *et al*. [6] demonstrated, this assumption might result in inaccurate localization of the lanes. For instance, unexpected driving behavior is likely to happen when an autonomous driving car encounters a mountainous road because the 2D planar geometry gives an erroneous perception of the 3D route.

The traditional methods for solving such problems cast perspective characteristics onto the BEV using camera intrinsic and extrinsic matrices [6, 7]. However, the feature map may not be ordered according to the perspective principle. Projecting a feature map from the front view to the top view can be unreasonable and detrimental to prediction performance. Moreover, for the aforementioned approaches, having multiple stages to do the task may not be convenient for real-time applications and complex scenarios.

To solve these problems, we decided to regress Lane position in 3D space while still working in a data-driven BEV projection approach like the Lift-Splat-Shoot (LSS) architecture [8] as see in Fig. 1. Our proposed network performs an end-to-end 3D lane regression instead of a two-stage approach like segmentation followed by lane modeling. Our approach has exceeded the performance of the previous state-of-the-art models [6, 7] in Average Precision (AP) and F-score metrics.

The contributions of this work are: (1) A Birds' Eye View (BEV)-based neural network architecture, similar to Lift-Splat-Shoot [8], but adapted to a 3D lane regression task; (2) A new loss function that exploits prior knowledge of the geometric constraints of parallelism in 3D space. Our contributions have shown significant improvements in AP, F-Score, and distance error metrics over state-of-the-art models such as 3D-LaneNet [6] and Gen-LaneNet [7], while our network doesn't make use of deep feature inverse perspective mapping.

II. Literature Review

3D lane recognition overcomes the planar road assumption and offers more accurate lane localization.

Manuscript received July 26, 2023; revised August 25, 2023; accepted September 21, 2023; published January 4, 2024.

However, 3D modeling using a monocular camera system is sensitive to variations in road slopes due to the absence of depth. As a workaround, many techniques employ constraints or rely on multi-sensor or multi-view camera setups [9–11].

Bai *et al.* [10] use camera and LiDAR sensors to detect lanes. However, LiDAR's practical application is limited by its cost and its data sparsity (e.g., the effective detection range is 48 m [10]). The authors of [9, 11] employed a less expensive stereo camera rig to carry out 3D lane detection, but they also experienced poor performance at higher distances. Some monocular techniques [6, 7, 12–14] use a single image and Inverse Perspective Mapping (IPM) of image-level deep features to forecast lanes in 3D space.

Ground-truth lanes and visual features are not aligned in the anchor representation [6], because 3D-LaneNet utilizes an unsuitable coordinate frame. This is most noticeable in the uphill road scenario when the parallel lanes projected to the virtual top-view appear nonparallel. The model was forced to acquire a global encoding of the entire scene when trained against such "dirty" ground-truth. As a result, the model had trouble generalizing to new scenarios that were different from the training distributions.

Gen-laneNet [7] offers a distinctive design in two ways. To directly generate genuine 3D lane points from the network output, it first provides a new geometry-guided lane anchor representation in a new coordinate frame. This highlights why a generalist approach to manage unfamiliar scenes matches the lane points with the underlying top-view features in the new coordinate frame. Gen-LaneNet also proposed a scalable two-stage approach that separates the learning of the geometry encoding sub-network from the learning of the image segmentation sub-network. However, the problem with this approach is that the feature maps might not be arranged in the new coordinate system in accordance with the inverse perspective principle, resulting in irrational projection and damaging the prediction accuracy.

We introduce the LS-3DLane architecture inspired by LSS [8], which, instead of explicit IPM, uses data and camera geometry-driven projection to regress lane position in 3D space from a single image in one-stage detection.

III. Materials and Methods

A. Problem Formulation

The goal of our targeted application, which involves receiving a 2D image from a camera mounted at the front of a moving vehicle, is to extract the 3D lane lines on the road, where each line consists of a group of N 3D points in the X, Y, Z vehicle coordinates. Our approach uses the data-driven projection method introduced in the Lift-Splat-Shoot (LSS) architecture [8] to project the camera-encoded features from the pixel coordinates to the Bird's Eye View (BEV) vehicle coordinates. The data-driven approach avoids the projection of the feature map from the front view to the BEV, as it can be unreasonable and detrimental to prediction accuracy, since the feature map may not be set up according to the perspective principle.

B. LS-3DLane

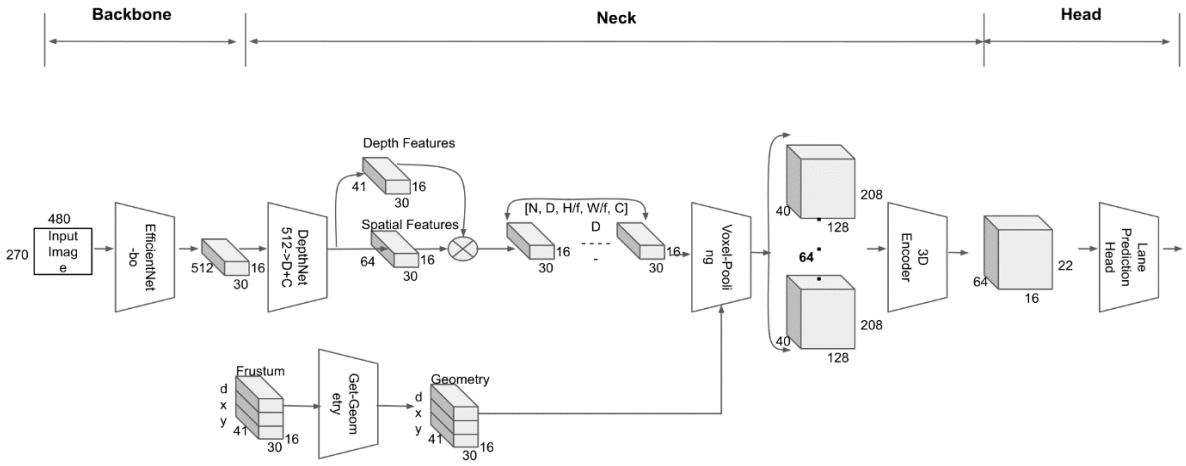
The LS-3DLane model, as shown in Fig. 1, is inspired by LSS [8]. It consists of three main parts. The first part is the Backbone, where the input image is processed by EfficientNet-b0 [15] to generate the encoded feature map. The second part is the Neck, which is a "Lift-Splat" pooling layer that lifts the feature map from the front view to the BEV. Our model differs from the LSS model [8] in that the feature map isn't collapsed in the z-dimension at the end of the voxel pooling layer, resulting in an output of 4D size 64 (number of channels) × 40 (z-dimension) ×208 (y-dimension) × 128 (x-dimension). This has improved the height detection of the lane points in the vehicle's Z-coordinate and improved the performance KPIs. The BEV encoder in LSS [8] was replaced by the 3D-encoder, which is a stack of 6 layers, each layer consisting of 3D-convolution, Batch-Normalisation, and RELU, where the first 3 layers are followed by 3D-Max-Pooling of stride 2. The third part is the Head, where the network predicts the 3D lane line in the virtual top-view space [7] in an anchor-based representation (see Fig. 2). The Lane prediction head architecture is shown in the bottom of Fig. 1.

C. Parallelism Regression Loss

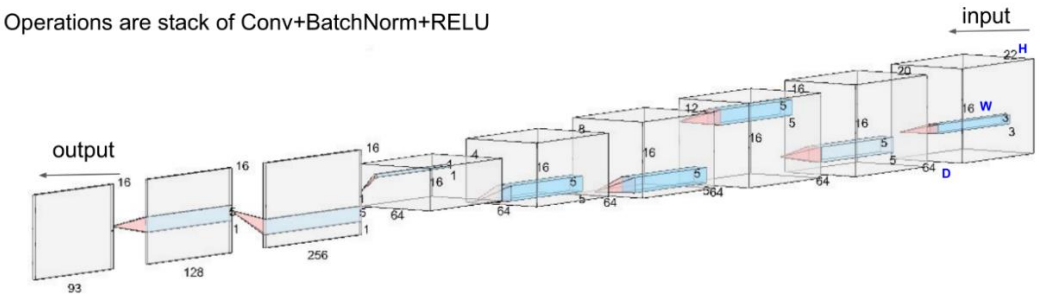
The training process is similar to that in Gen-LaneNet [7], where the input image and its associated ground-truth 3D lane labels are used. Each lane curvature in the ground truth is projected to the virtual top-view [7] and associated with the nearest anchor at Y_{ref} (see Fig. 2). Based on the ground truth values at the predefined y-positions $\{y_i\}_j^K$, the ground truth anchor attributes are computed. The loss function is calculated between the predicted anchors X_i and the ground truth anchor $\hat{X}_i = \{(\hat{x}_i^t,\ \hat{z}_i^t,\ \hat{v}_i^t,\ \hat{P}_i^t)\}^{t\in\{c,l\}}$,

$$
\begin{aligned}
l = &- \sum_{t\in\{c,l\}} \sum_{i=1}^{N} (P_i^t \log P_i^t + (1-P_i^t)\log(1-P_i^t)) \\
&- \sum_{t\in\{c,l\}} \sum_{i=1}^{N} (\hat{v}_i^t \log v_i^t + (1-\hat{v}_i^t)\log(1-v_i^t)) / K \\
&- \sum_{t\in\{c,l\}} \sum_{i=1}^{N} (P_i^t.(\| \hat{v}_i^t.(x_i^t - x_i^t) \|_1 + \| \hat{v}_i^t.(z_i^t - \hat{z}_i^t) \|_1)
\end{aligned}
\tag{1}
$$

where $\hat{x}_i^t$ is the x-offset value, $\hat{z}_i^t$ is the z-coordinate value and $\hat{v}_i^t$ is the visibility value at each y_i and $\hat{P}_i^t$ is 1 if the current anchor is associated to ground truth lane line or 0 otherwise and t is either a type centerline or laneline.



(a)



(b)

Fig. 1. Schematic of the Network architecture. (a) is the LS-3DLane inspired by the LSS architecture [8] but using only one camera input instead of 6 cameras. (b) is the lane prediction head architecture, a stack of Convolution+Batch-Normalisation+RELU layers.

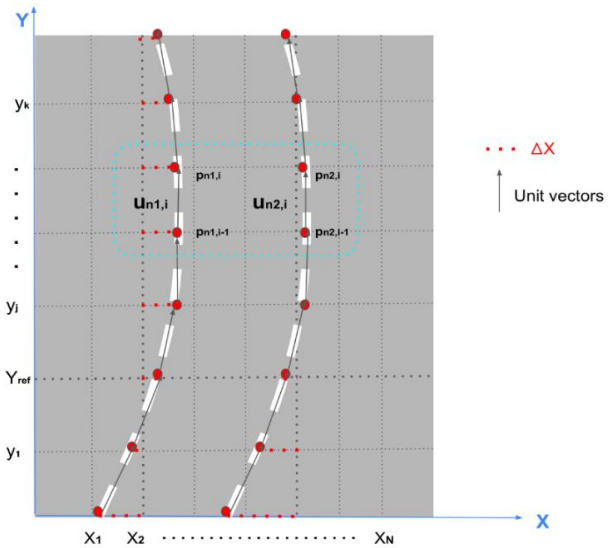


Fig. 2. Geometric representation of anchors. Dashed white lines are the ground lane markings. The parallelism loss is calculated between the unit vectors pairs as in the cyan rectangle. Red points are the lane points calculated at the predicted anchor in the 3D vehicle coordinates.

By exploiting prior knowledge of the geometric structure of the lanes, i.e., that most samples are parallel lanes on the highway, we can penalize non-parallel predicted lanes during model training through a parallelism loss $l_{\text{parallelism}}$ to improve performance. As shown in Fig. 2, the predicted lane points are calculated in the 3D vehicle coordinates $\in \{x_{3D}, y_{3D}, z\}_i$ after being transformed from the virtual top view [7] to satisfy the parallelism assumption ex. The predicted lane points $p_{n_1,i}$, $p_{n_1,i-1}$,$p_{n_2,i}$ and $p_{n_2,i-1}$ (see in Fig. 2), where n_1 and n_2 are the anchor line indices. Unit vector is calculated between each two successive points in the lane ex. $\hat{u}_{n,i} = \frac{p_{n,i}-p_{n,i-1}}{||p_{n,i}-p_{n,i-1}||}$. The unit vector is visible if both points constructing it are visible ex. $\widehat{v\_vect}_{i,n} = \hat{v}_{n,i} \times \hat{v}_{n,i-1}$. The dot product is calculated between each corresponding

vectors as in the cyan rectangle (see in Fig. 2) then subtracted from 1, ex. $||1-\hat{u}_{n_1,i}.\hat{u}_{n_2,i}||_1$ and summed over the whole lane. This is only applied for valid anchors ex. $\hat{P}_n^t = 1$. The parallelism loss is calculated as follow

$$l_{parallelism} = \sum_{t\in\{c,l\}} \sum_{\substack{n_1=1 \\ P_{n_1}^t=1}}^{N-1} \sum_{\substack{n_2=n_1+1 \\ P_{n_2}^t=1}}^{N} \left[\sum_{i=1}^{K-1} \left[v\_vect_{i,n_1}^t \times v\_vect_{i,n_2}^t \times \|1-u_{n_1,i}.u_{n_2,i}\|_1 \right] \right], \quad where\ P_{n_2}^t = 1 \tag{2}$$

The total training loss is $l_{total} = l(eq.1) + l_{parallelism}$.

IV. Experiments and Results

All experiments were conducted using an Nvidia TITAN-X (Pascal) GPU equipped with 12 GB of dedicated memory.

A. Datasets

Our network was trained on a synthetic dataset for 3D lane detection [16] containing over 10,000 images. This dataset was created to encourage the development and evaluation of 3D lane identification techniques and is an addition to the synthetic dataset created by Apollo. The full building approach and evaluation method are described in the ECCV 2020 report [7]. For training, we used the visual variation subset consisting of 3,968 training images and 472 validation images. The original input image resolution is 1080 (H) × 1920 (W), which was resized to 270 × 480 and randomly rotated by an angle sampled from the uniform distribution U(−10º, 10º). The initial top-view layer should have the same spatial resolution of 208×128 to represent a flat area with a range of [−32, 32] [1, 104] m along the x and y axes, respectively, with a resolution of 0.5 m/cell.

B. Evaluation Metrics

The evaluation metrics are the same as those used in GenLaneNet [7], namely AP, F-score, and spatial errors in the X, Y, and Z directions.

C. Training Configuration

The Adam optimizer was used with a learning rate of $1e^{-3}$ and weight decay of $1e^{-7}$. The batch size was set to 6.

D. Results

The 3D-LaneNet [6] and the Gen-LaneNet [7] models were re-trained using the resized input size of 270×480. To evaluate the performance of our LS-3DLane network, we tested it on the validation data of three different subsets: Balanced Scenes, Visual Variation, and Rarely Observed Data. The Balanced Scenes subset contains images taken in clear daytime lighting with no light variations, while the Visual Variation subset simulates light variation due to different weather conditions, and the Rarely Observed Data subset replicates severe weather and light condition scenes.

The results of our evaluation are reported and compared to the performance of the 3D-LaneNet and Gen-LaneNet models in Table I. All the models are trained with APOLLO 3D lane synthetic dataset using an input image size of 270×480. The AP and the F-score KPIs are in percentage ratio % and the X and Z errors are in meters. The "Close" error means the average error before 40 meters range and the "Far" error means the average error from 40 meters to 100 meters ranges. Our model outperforms both models by a significant margin of 10.4% and 5.7% in terms of AP score, respectively. Additionally, our model shows fewer falsely classified predicted lines as lanes and demonstrates 3.8% and 2.2% better F-Scores, respectively. The improvement in the F-Scores indicates that our model achieves a better balance between the precision metric and the recall metric by reducing the number of ground truth lanes ignored in the prediction. These results suggest that our LS-3DLane network is better at predicting the lane geometrical structure in 3D space and is more accurate in matching the ground truth compared to the other models.

TABLE I. This Table Compares the Results of the LS-3DLane to the Gen-LaneNet [7] and the 3D-LaneNet [6]

Scene	Method	AP	F-Score	X error (Close)	X error (Far)	Z error (Close)	Z error (Far)
Balanced Scenes	**LS-3DLane**	89.6	89.4	0.0834	0.480	0.0217	0.271
	GenLanetNet	83.9	84.9	0.0704	0.578	0.0152	0.249
	3D-LaneNet	79.2	85.2	0.0651	0.516	0.0185	0.229
Rarely Observed	**LS-3DLane**	79.9	78.1	0.173	0.952	0.0469	0.705
	GenLanetNet	69.4	72.6	0.158	1.020	0.0386	0.632
	3D-LaneNet	56.9	68.0	0.165	0.969	0.0468	0.596
Visual Variations	**LS-3DLane**	89.0	87.3	0.0848	0.529	0.0321	0.330
	GenLanetNet	84.6	85.1	0.0705	0.590	0.0152	0.272
	3D-LaneNet	76.5	83.5	0.0726	0.534	0.0209	0.252

We further evaluated our model's performance by measuring the error metrics in the X and Z vehicle coordinates of the predicted lane position. These metrics measure the average distance error between the predicted lane and its corresponding ground truth lane in the X and Z directions. We measured both metrics at close range (5 m to 40 m) and far range (40 m to 100 m) in the driving direction (Y-vehicle coordinate). Our model demonstrated comparable results in terms of the X/Z errors, further confirming the accuracy of our model's predictions.

To illustrate the effectiveness of our approach, we present an example of our model's prediction performance on an elevating road with a curvature of 4 models in Fig. 3. As evident from the example, our model performed better than the previous models, highlighting the superiority of our approach in detecting lanes accurately.

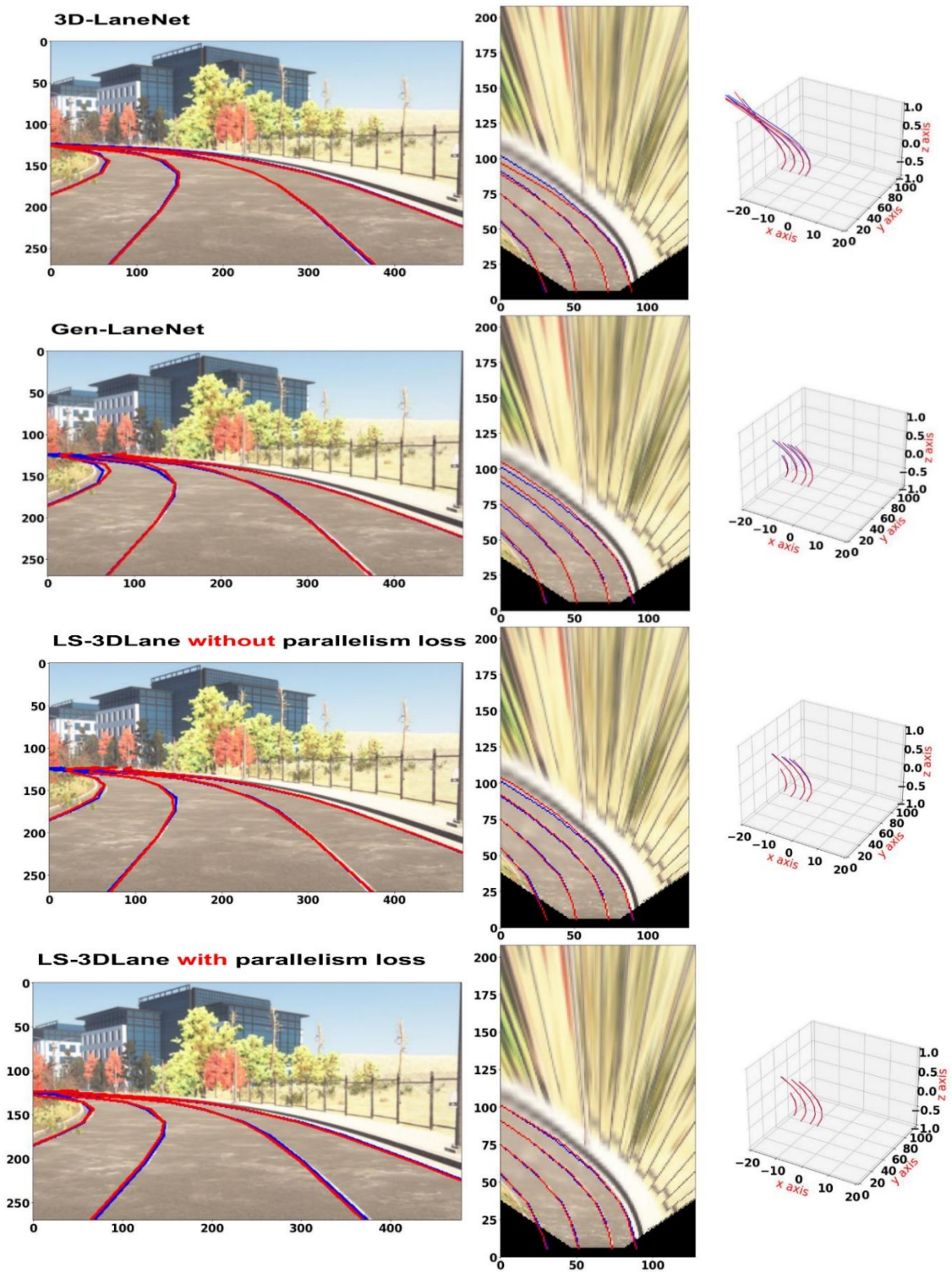


Fig. 3. Example of an elevating road with a curvature. Blue lines are the ground truth, Red lines are the predicted lanes. (Left) Front image, (Middle) BEV projection, (Right) 3D Space. Such example shows a clear advantage of our model over the 3D-LaneNet and the Gen-LaneNet approaches and the added value of applying the Parallelism loss while training.

E. Parallelism Loss Effect

We conducted an ablation study in Table II, where the parallelism regression loss was dropped from the total training loss, and the model was retrained. In Table II, "w/o" represents the drop of the parallelism loss while training and "w" represents the inclusion of the parallelism loss. The KPI results are reported in Table II, showing the

gains in the performance of the original model when applying the parallelism loss while training the model. We also integrated the parallelism training loss while retraining the 3D-LaneNet [6] and the Gen-LaneNet [7] models to test if this positive effect can be applied to other models. As seen in Table II, most of the time, applying the parallelism loss in addition to the total loss has improved the performance, with gains that can reach 4.5% in AP score and 2.8% in the F-score.

Fig. 3 shows an example of the added value to the lane prediction performance when applying the parallelism loss during training (third and fourth images).

TABLE II. COMPARISON OF THE PARALLELISM LOSS EFFECT

		Balanced Scenes			**Rarely Observed**			**Visual Variations**		
		w/o	w	**gain**	w/o	w	**gain**	w/o	w	**gain**
LS-3DLane	AP	87.1	89.6	+2.5	75.7	79.9	+4.2	84.9	89	+4.1
	F-Score	88.2	89.4	+1.2	76.1	78.1	+2	85.4	87.3	+1.9
	X-error close	0.0975	0.0834	−0.0141	0.175	0.173	−0.002	0.101	0.0848	−0.0162
	X-error far	0.539	0.48	−0.059	0.992	0.952	−0.04	0.595	0.529	−0.066
	Z-error close	0.0234	0.0217	−0.0017	0.0529	0.0469	−0.006	0.0339	0.0321	−0.0018
	Z-error far	0.273	0.271	−0.002	0.704	0.705	+0.001	0.324	0.33	+0.006
Gen-LaneNet	AP	83.9	85.3	+1.4	69.4	73.9	+4.5	84.6	86.2	+1.6
	F-Score	84.9	85.7	+0.8	72.6	75.4	+2.8	85.1	85.7	+0.6
	X-error close	0.0704	0.0655	−0.0049	0.158	0.144	−0.014	0.0705	0.0655	−0.005
	X-error far	0.578	0.567	−0.011	1.02	0.999	−0.021	0.59	0.592	+0.002
	Z-error close	0.0152	0.0148	−0.0004	0.0386	0.0389	+0.0003	0.0152	0.0144	−0.0008
	Z-error far	0.249	0.255	+0.006	0.632	0.649	+0.017	0.272	0.274	+0.002
3D-LaneNet	AP	79.2	80.7	+1.5	56.9	58.2	+1.3	76.5	76.4	−0.1
	F-Score	85.2	86.7	+1.5	68	69.6	+1.6	83.5	83.4	−0.1
	X-error close	0.0651	0.0591	−0.006	0.165	0.14	−0.025	0.0726	0.0655	−0.0071
	X-error far	0.516	0.452	−0.064	0.969	0.883	−0.086	0.534	0.491	−0.043
	Z-error close	0.0185	0.0152	−0.0033	0.0468	0.0428	−0.004	0.0209	0.0177	−0.0032
	Z-error far	0.229	0.219	−0.01	0.596	0.589	−0.007	0.252	0.247	−0.005

V. CONCLUSIONS

This work is a proof of concept that demonstrates the use of the Lift-Splat LSS architecture for lane position detection in 3D space using a monocular camera. We modified the LSS architecture to predict lane height, which allowed for projection to 3D space in a data-driven manner rather than relying solely on the IPM. Our proposed end-to-end network performed well for all KPIs compared to previous approaches. We also introduced the parallelism loss, which leverages prior knowledge of the geometrical structure of lanes. This loss can be applied to any anchor-based 3D lane detection network and significantly boost performance.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Mohammed Hassoubah conducted the literature review, proposed and implemented the novel solution of the parallelism loss function, conducted the experiments, and wrote the paper. Ganesh Sistu provided guidance and supervision to Mohammed Hassoubah in the area of lane detection, and also reviewed the paper before submission. All authors have approved the final version.

REFERENCES

[1] Commaai. (2023). Openpilot. [Online]. Available: https://github.com/commaai/openpilot

[2] L. Liu, X. Chen, S. Zhu, and P. Tan, "Condlanenet: A top-to-down lane detection framework based on conditional convolution," in *Proc. the IEEE/CVF international Conference on Computer Vision*, 2021, pp. 3773–3782.

[3] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on local: Detecting lane marker from bottom up via key point," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14122–14130.

[4] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[5] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. 2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 286–291.

[6] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3d-lanenet: End-to-end 3d multiple lane detection," in *Proc. the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2921–2930.

[7] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, and T. E. Choe, "Gen-LANENET: A generalized and scalable approach for 3d lane detection," in *Proc. Computer Vision—ECCV 2020: 16th European Conference*, Springer, 2020, pp. 666–681.

[8] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proc. Computer Vision—ECCV 2020: 16th European Conference*, Springer, 2020, pp. 194–210.

[9] P. Coulombeau and C. Laurgeau, "Vehicle yaw, pitch, roll and 3d lane shape recovery by vision," in *Proc. Intelligent Vehicle Symposium*, IEEE, 2002, vol. 2, pp. 619–625.

[10] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun, "Deep multi-sensor lane detection," in *Proc. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3102–3109.

[11] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, and C. Pocol, "3D lane detection system based on stereovision,"in *Proc. the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, 2004, pp. 161–166.

[12] R. Liu, D. Chen, T. Liu, Z. Xiong, and Z. Yuan, "Learning to predict 3d lane shape and camera pose from a single image via geometry constraints," in *Proc. the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, no. 2, pp. 1765–1772.

[13] F. Yan, M. Nie, X. Cai, J. Han, H. Xu, Z. Yang, C. Ye, Y. Fu, M. B. Mi, and L. Zhang, "Once-3dlanes: Building monocular 3d lane detection," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17143–17152.

[14] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao *et al.*, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," in *Proc. Computer Vision–ECCV 2022: 17th European Conference*, Springer, 2022, pp. 550–567.

[15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning*, 2019, pp. 6105–6114.

[16] A synthetic dataset for 3d lane detection. [Online]. Available: https://github.com/yuliangguo/3D Lane Synthetic Dataset