3D-SpLineNet: 3D Traffic Line Detection using Parametric Spline Representations

Maximilian Pittner[1,2], Alexandru Condurache[1,2], Joel Janai[1]
[1]Bosch Mobility Solutions, Robert Bosch GmbH, 71229 Leonberg, Germany
[2]Institute of Signal Processing, University of Lübeck, 23562 Lübeck, Germany
{Maximilian.Pittner, AlexandruPaul.Condurache, Joel.Janai}@de.bosch.com

Abstract

Monocular 3D traffic line detection jointly tackles the detection of lane markings and regression of their 3D location. The greatest challenge is the exact estimation of various line shapes in the world, which highly depends on the chosen representation. While anchor-based and grid-based line representations have been proposed, all suffer from the same limitation, the necessity of discretizing the 3D space. To address this limitation, we present an anchor-free parametric lane representation, which defines traffic lines as continuous curves in 3D space. Choosing splines as our representation, we show their superiority over polynomials of different degrees that were proposed in previous 2D lane detection approaches. Our continuous representation allows us to model even complex lane shapes at any position in the 3D space, while implicitly enforcing smoothness constraints. Our model is validated on a synthetic 3D lane dataset including a variety of scenes in terms of complexity of road shape and illumination. We outperform the state-of-the-art in nearly all geometric performance metrics and achieve a great leap in the detection rate. In contrast to discrete representations, our parametric model requires no post-processing achieving highest processing speed. Additionally, we provide a thorough analysis over different parametric representations for 3D lane detection. The code and trained models are available on our project website https://3d-splinenet.github.io/.

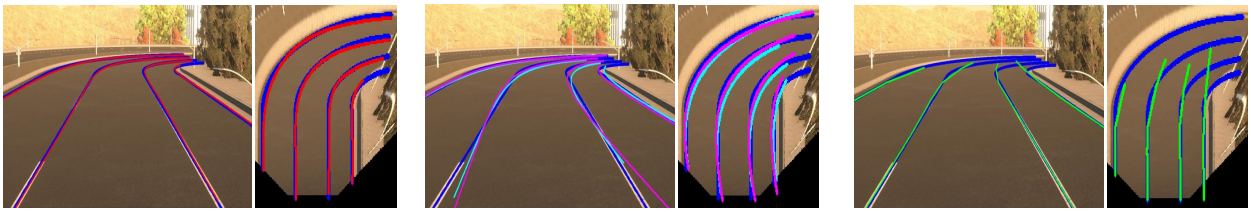
1. Introduction

Traffic line detection is a fundamental part of driver assistance systems and autonomous driving. Such systems have to estimate the accurate location of lane markings in the 3D world to realize a safe driving behavior. The task is often formulated as a monocular detection problem using a front-facing camera as primary sensor.

One common strategy is to directly detect the lane markings in the 2D image and afterwards project them into the 3D world. Classical methods [1, 5, 19, 17, 44, 41] apply hand-crafted filters to extract local features like edges to localize line segments and cluster them in a post-processing step. These rule-based algorithms assume a certain appearance of lanes and, therefore, fail for more complex examples. Consequently, Convolutional Neural Networks (CNNs), which are capable of capturing global context, have been proposed to extract road markings using pixel representations based on segmentations [24, 34, 12, 15] and geometrical representations based on straight-line anchors [25, 43, 42] or grids that model lane geometry in each cell [18, 22]. While these discrete representations require a subsequent curve fitting step to sufficiently describe complete line objects, parametric methods [10, 46, 26, 29, 9] directly model lines as continuous functions in the image and let the network predict the required parameters. This spares the necessity of post-processing and allows for more flexible modeling of lane geometry.

Finally, the transformation of resulting 2D detections into the 3D world is commonly performed by means of a homography assuming a flat road plane, due to the lack of depth information. Since this assumption is frequently violated, more sophisticated methods for road surface estimation [47] are necessary and, eventually, 2D lane marking detections have been used with the assumption of parallelism between traffic lines to address this problem [48].

This has led to monocular 3D lane detection methods [11, 13, 8] that jointly tackle the lane detection and 3D estimation problem while leveraging the commonalities of both problems. In general, features are extracted from the input image, projected using inverse perspective mapping (IPM) into top-view and provided to anchor- [11, 13] or grid-based [8] detectors to reconstruct lane markings. However, these methods suffer from the usual drawbacks of discrete representations. Anchor-based methods face problems for complex shapes deviating strongly from the underlying assumption and use interpolated ground truth values for training, which results in additional errors. Grid-based methods, on



(a) Our Splines (b) Polynomials (c) Anchor-based [13]

Figure 1: Traffic line predictions with *3rd degree Splines*, *3rd degree* and *5th degree* polynomials, *anchors* resulting from different representations compared to the *ground truth*.

the other hand, need a high amount of parameters to achieve sufficient resolutions and require clustering to reconstruct full lines from grid cells.

Inspired by 2D lane detection methods [10, 46, 26], we propose a parametric representation describing lines as continuous curves in 3D space to overcome these limitations. We show that previous parametric 2D approaches fail to achieve SOTA performance due to the utilization of too simplistic line models. Thus, we suggest a more sophisticated representation based on B-Splines, which is capable of sufficiently capturing complex line shapes and at the same time requires smaller amount of parameters. Performing parametric regression in continuous space pays the same attention to lanes of various geometries appearing at any position in space and does not depend on the choice of pre-defined locations. We are able to learn the parameters directly from the ground truth without any usual pre-processing such as line fitting or interpolation. Additionally, our ground truth association ensures optimal matching of line proposals and considers all lines appearing in the image. Fig. 1 illustrates the advantages of our splines over polynomials and anchor-based representations even in simple situations.
Our main contributions can be summarized as follows:

- We propose an end-to-end trainable architecture that directly predicts parameters describing continuous traffic lines in 3D from monocular images and present a way to train it using regression in continuous space.
- We present a better strategy for association of ground truth markings to 3D line candidates based on the reduction of the mean distance.
- We compare different parametric representations in a thorough analysis.
- Our method achieves state-of-the-art performance on a synthetic dataset and greatly improves the detection rate and runtime in comparison to previous methods.

2. Related work

The reliable detection of road markings from video-based input has been investigated for almost four decades [7, 20] and was already an essential part of the first autonomous driving projects (e.g. PROMETHEUS [6]). Apart from classical methods [2, 31], deep neural networks have gained attention in recent years to address the problem. By learning global context information, they were able to outperform classical methods especially in challenging situations (e.g. bad visibility, occlusions, etc.) [45].

2.1. Representations in 2D traffic line detection

In the meantime, different representations have been proposed to address 2D traffic line detection with deep neural networks. Early methods have formulated the problem as a segmentation task striving for classifying traffic lines on *pixel-level* [24, 34, 12, 33, 15, 35, 49]. Other approaches suggest to reconstruct lines from a coarse *grid representation* [22, 18] instead of high-resolution segmentation by performing regression on local line segments [18] or keypoints [22] per cell. However, both representations require a subsequent clustering step to distinguish multiple line instances, e.g. using learned embedding vectors.

While these representations barely make assumptions about line geometry, *anchor-based approaches* [25, 43, 42], inspired by famous object detectors [38, 37, 27], describe traffic lines as straight line anchors with deviations at predefined locations. Thus, they aim for classifying the most suitable anchor and learning positional offsets by regression. While Line-CNN [25] uses straight anchors of different orientations, related methods vary in the design of anchors and introduce attention mechanism [43] or incorporate structural information like line parallelism [42].

Typically, all mentioned discrete representations require a post-processing step to fit a smooth curve to the detected discrete points or segments. *Parametric representations* [10, 46, 26, 29, 9], by contrast, directly model lines as continuous functions and obtain the function parameters from the network. While in [10] least-squares fitting was applied to feature maps to obtain polynomial coefficients, PolyLaneNet [46] directly predicts these parameters and [26] uses a similar approach but employs the powerful transformer architecture as a backbone. In concurrent development to our work, [9] was presented, which uses Bézier

curves as a parametric line model, which were utilized in other applications like scene text detection [28] before. Our line representation is inspired by [46] and [26] but instead of describing lines as one-dimensional functions, we propose to model them as continuous parameterized 3D curves in a world coordinate system. Observing that polynomials are limited in their representation capabilities, we do not constrain our network to one representation and also consider B-Splines [4, 40] that are more capable to model 3D lane geometry and have been used successfully for classical road surface estimation [47, 48]. In contrast to polynomials and Bézier curves, B-Splines benefit from independent basis functions, which is particularly advantageous for modeling typical road shapes (see Fig. 1).

2.2. 3D traffic line detection

One way to obtain world coordinates from 2D lane detection is to assume a flat road plane and project lines to this plane using a homography. Many methods directly detect lanes in a virtual top-view, which can be constructed using inverse perspective mapping [30]. IPM has been used in classical traffic line detection [36] and also in deep learning based 2D methods [33, 14, 29], where the top-view serves as input to the network detecting traffic lines directly. Since lines modeled in top-view only hold meaningful information under the flat road assumption, methods have been proposed to overcome this limitation. Classical 3D traffic line detection approaches make hard assumptions about the road model [3, 1] use stereo vision [32] or multi-sensor data [16] to solve the problem of depth ambiguity. Eventually, it was shown that 2D detection methods already provide enough information for road surface estimation [48] when leveraging the parallelism of traffic lines.

Monocular 3D traffic line detection methods [11, 13, 8] were thus proposed that learn 3D geometry directly on the basis of images from 3D ground truth in a supervised manner. While these approaches also use IPM, they learn deviations from the hypothetical flat road plane, i.e. revealing the vertical height component of the 3D traffic line geometry. In contrast to the front-view, the top-view serves as a reasonable input for estimating height, since parallel lines appear diverging in uphill scenes and converging in downhill scenes. 3D-LaneNet [11] proposes an end-to-end trainable dual-way architecture, where one pathway extracts feature maps of different scales and transforms them to the top-view using IPM. The second pathway processes these features to predict traffic lines using an anchor-based representation formed by straight lines and positional offsets in lateral direction and height. Gen-LaneNet [13] exploits anchors similarly, but uses a geometric transformation that aligns the predictions to the top-view. They also suggest a two-stage architecture, which consists of a backbone trained on binary line segmentation and a detection head operating on the resulting top-view segmentation map. 3D-LaneNet+ [8] is based on a similar architecture as [11] but uses a grid-representation as output to estimate local 3D line parameters per cell and learn embeddings to cluster cells subsequently. Unfortunately, the latter does not provide trained models, a code base or an evaluation scheme for reproduction of results and comparison.

Regarding the network architecture, our method is related to Gen-LaneNet, but we suggest to train the whole framework end-to-end and feed multi-channel features through the detection head instead of binary segmentation maps. More important, we avoid a discrete representation based on anchors or grids. Instead, we propose a parametric formulation to model traffic lines as 3D curves, where the lateral and vertical components are described by B-Splines. Our representation allows us to model complex line shapes and learn from real ground truth in continuous space. Besides, our method achieves high speed as it does not require costly post-processing such as line fitting or clustering.

3. Methodology

The following section describes our 3D traffic line detection approach. The main focus lies on our novel parametric 3D line representation and our proposed training scheme. An overview of our method is illustrated in Fig. 2.

3.1. Traffic line representation

Inspired by prior work in 2D traffic line detection [10, 46, 26, 9], our approach uses a parametric representation to model lines as continuous curves. Contrary to these methods, where a single function suffices for the 2D geometry in a plane, we intend to describe 3D geometry. Consequently, lines are represented as parameterized 3D curves as

$$\boldsymbol{l}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \boldsymbol{\eta} \odot \boldsymbol{f_l}(t) = \boldsymbol{\eta} \odot \begin{pmatrix} f_{\boldsymbol{l}_x}(t) \\ t \\ f_{\boldsymbol{l}_z}(t) \end{pmatrix} , \quad (1)$$

with normalization vector $\boldsymbol{\eta} \in \mathbb{R}^3$, continuous vector function $\boldsymbol{f_l} : \mathbb{R} \rightarrow \mathbb{R}^3$, curve argument $t \in [t_s,\ t_e]$, where $t_s,\ t_e \in [0, 1]$, and $\odot$ the element-wise product.

As is common practice, the origin and orientation of the 3D reference frame is defined by the camera pitch angle and height, which we consider given. Hence, the x-y-plane corresponds to the top-view projection plane of IPM (see Fig. 2). We introduce $\boldsymbol{f_l}$ describing the shape of 3D curves in a normalized space and the vector of normalization constants $\boldsymbol{\eta} = [\eta_x,\ \eta_y,\ \eta_z]^T$ for rescaling each dimension. Generally, $\boldsymbol{f_l}$ can be modeled by any kind of continuous function. A minor simplification is to only use elaborate functions for $x(t)$ and $z(t)$ to model the lateral and vertical deflections of the ego-direction $y(t)$. Since lines usually appear with different ranges, we also need to model the start

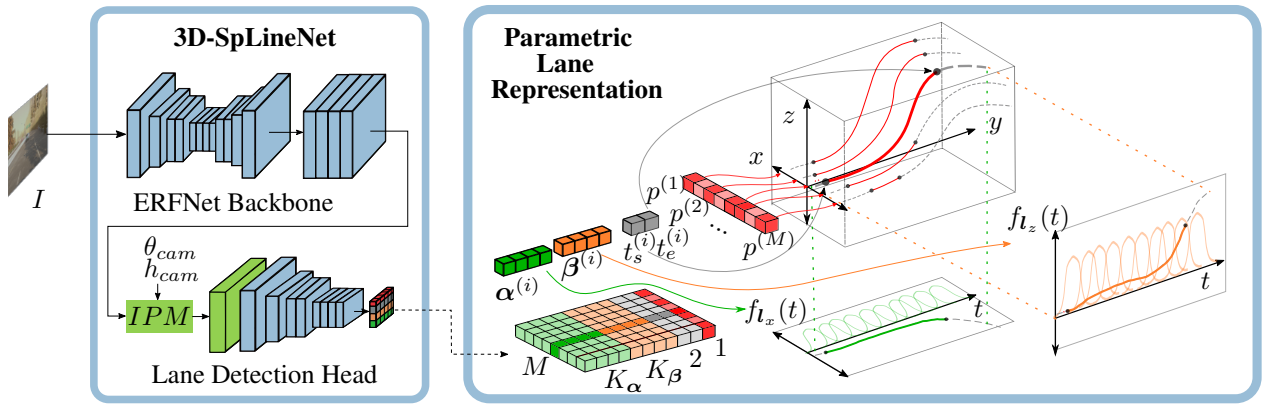


Figure 2: An overview of our proposed 3D traffic line detection framework. *3D-SpLineNet* receives a front-view RGB image I, feeds it through the *Backbone* and processes top-view feature maps by the *Lane Detection Head*. The final output contains a set of parameters for M 3D curve proposals, namely $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ for line shape, t_s, t_e for line range and existence probability p .

and end point of lane markings. We formulate this with a fixed interval $t \in [t_s,\ t_e]$, where $\eta_y \cdot t_s$ and $\eta_y \cdot t_e$ define the start and end of the lane in y-direction.

As discussed in Section 2.1, polynomial functions have already been used in 2D approaches [10, 46, 26, 29]. However, high degrees are necessary to accurately describe even simple courses of lane markings as illustrated in Fig. 1b. In contrast, B-Splines [4, 40] are piecewise polynomial functions that can also represent complex courses of lane markings due to their piecewise definition and the independence among basis functions. Consequently, compared to polynomials, lower degrees are sufficient to model typical road shapes as shown in Fig. 1.

Thus, we use B-Splines to model the lateral and vertical components with additional offsets, such that we obtain

$$\boldsymbol{f_l}(t) = \begin{pmatrix} \sum_{k=1}^{K_B} \alpha_k \cdot B_{k,d}(t) + \alpha_0 \\ t \\ \sum_{k=1}^{K_B} \beta_k \cdot B_{k,d}(t) + \beta_0 \end{pmatrix}, \qquad (2)$$

where each of K_B basis functions $B_{k,d}(t)$ represents a piecewise polynomial of degree d and covers a certain domain defined by a set of knots $\{t_1, t_2, \ldots, t_{K_B+1-d}\}$. $\{\alpha_k, \beta_k\}_{k=1}^{K_B}$ is the set of control points specifying the impact of each basis function, i.e. controlling the shape of the curve $\boldsymbol{f_l}(t)$. α_0, β_0 are offsets to model mean shifts.

Since the proposed algorithm processes images captured by a single forward-facing camera, observable lines usually progress monotonously in driving direction. Hence, for the detection of relevant lanes it is sufficient to use a curve model that parameterizes the driving direction $y(t)$ as a scaling of the curve argument t. This model only shows limitations in cases of horizontal lanes and special scenarios like U-turns or roundabouts, but is still capable of representing almost horizontal lines, junctions and steep curves. In future, when focusing on applications in urban environments, a conceivable extension is to additionally parameterize $y(t)$ with splines. More details on our representation and possible lane geometries are provided in the supplementary.

3.2. Network architecture

Inspired by Gen-LaneNet [13], we also use a semantic segmentation backbone to extract information from the front-view and a lane detection head, but make meaningful modifications. Gen-LaneNet [13] addressed the problem of limited 3D data using a fixed pre-trained backbone providing binary lane masks to the detection head in order to decouple the 2D segmentation from the 3D geometry estimation task. Since our emphasis does not lie on such cases, we propose to directly use the features of the backbone instead and train the whole architecture end-to-end. For this, we replace the last layer such that we obtain a multi-channel feature map, project it to the top-view and feed it through the lane detection head. For the top-view transformation we apply IPM [30] as proposed in [11, 13, 8]. Training backbone and lane detection head end-to-end allows the backbone to learn richer feature maps for the 3D estimation task and our detection head can leverage the full backbone capacity.

The final layer of our detection head is of size $M \times (K_{\boldsymbol{\alpha}} + K_{\boldsymbol{\beta}} + 3)$. It consists of M proposals, where each includes $K_{\boldsymbol{\alpha}}$ and $K_{\boldsymbol{\beta}}$ parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as control points and offsets for the x- and z-component, two parameters for start and end and one probability p that a line exists for the proposal. Note that using the proposed B-Spline representation from Eq. (2) yields $K_{\boldsymbol{\alpha}} = K_{\boldsymbol{\beta}} = K_B + 1$ shape parameters. Finally, the overall network output is given as $\left\{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, t_s^{(i)}, t_e^{(i)}, p^{(i)}\right\}_{i=1}^{M}$. Fig. 2 shows an overview of our end-to-end trainable network architecture and our parametric representation.

3.3. Training

Initialization. In 3D space, traffic lines can occur with various geometries (e.g. left or right curves, up- and downhill, different lengths). Hence, covering the huge variety of lane appearance with adequate initializations would lead to a high amount of proposals. Assuming the majority of lanes contain straight segments progressing in driving direction, we restrict the number of proposals to a feasible amount and use straight line initializations uniformly distributed along normalized x-direction in the range of $[-0.5,\ 0.5]$.

Association to ground truth. Contrary to 3D-LaneNet [11] and Gen-LaneNet [13], which use a fixed reference point, we suggest to consider the mean lateral distance from the ground truth lines to the line proposals as a matching criterion. This allows us to associate all lane markings, also those not passing a specific y-position. However, in typical road scenes the first segment of most traffic lines is very close to our straight line initialization. Thus, instead of considering the mean lateral distance over an entire line, we suggest to only consider a specific ratio of the ground truth. In this way, the network can benefit from appropriate initializations, which we investigate in Section 4.2. Even with such an association strategy a unique assignment of the ground truth to candidates cannot be ensured, e.g. in case of multi-markings. We resolve this problem using the Hungarian matching [23] for the association. Fig. 3 shows (highlighted in red) the main differences between our and previous assignment strategies.

Supervised detection loss. Our objective to train the network on the detection task includes a classification loss $\mathcal{L}_c$ to learn line presence, a shape loss $\mathcal{L}_s$ to minimize the distance of each line instance to the ground truth, and a range loss $\mathcal{L}_r$ to learn the start and end of the line range. For the classification loss the common binary cross-entropy is used, such that we obtain

$$\mathcal{L}_c = -\sum_{i=1}^{M} \hat{p}^{(i)} \log p^{(i)} + (1-\hat{p}^{(i)}) \log(1-p^{(i)}), \quad (3)$$

with binary labels $\hat{\boldsymbol{p}}$ indicating the presence (association) of ground truth lines.

To learn line shapes, we propose a parametric regression formulation that minimizes the L_1-distance between two 3D curves. For a predicted line instance $\boldsymbol{l}(t)$ and its corresponding ground truth $\hat{\boldsymbol{l}}(t)$ we obtain

$$\mathcal{L}_s = \int_{\hat{t}_s}^{\hat{t}_e} \left\| \boldsymbol{w}(t) \odot \left(\boldsymbol{f_l}(t) - \boldsymbol{\eta}^{-1} \odot \hat{\boldsymbol{l}}(t) \right) \right\|_1 dt \quad (4)$$

$$= \int_{\hat{t}_s}^{\hat{t}_e} \left(w_x(t) \cdot \left| f_{l_x}(t) - \frac{1}{\eta_x} \hat{x}(t) \right| + \right. \quad (5)$$

$$\left. w_z(t) \cdot \left| f_{l_z}(t) - \frac{1}{\eta_z} \hat{z}(t) \right| \right) dt, \quad (6)$$

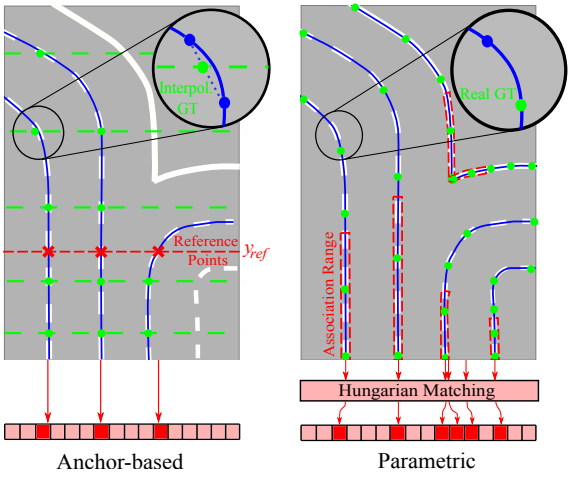


Figure 3: Comparison of anchor-based and our parametric approach. Red: Our association considers all lines while the reference point matching misses lines not passing y_{ref}. Green: Our method performs regression over the entire line using real ground truth. Anchor-based methods only learn deviations at pre-defined locations requiring interpolation.

where $\hat{t}_s$, $\hat{t}_e$ are the start and end of the ground truth lane. $\boldsymbol{w}(t)$ is a weighting function, which considers the standard deviations of lane geometry. Thus, it enables us to treat near- and far-range errors well-balanced in the regression loss. More details on $\boldsymbol{w}(t)$ are provided in the supplementary. In practice, the integral is approximated numerically by choosing an appropriate amount of annotated ground truth points uniformly distributed along the line and computing the point-wise distance to the associated predictions. The predicted values for x and z are obtained by recovering t-values from the ground truth ($\frac{\hat{y}}{\eta_y}$) and evaluating the function components of $f_{l_x}(t)$ and $f_{l_z}(t)$.

For learning the line range, we use a simple regression of the start and end point $\hat{t}_s$, $\hat{t}_e$ of the ground truth lane

$$\mathcal{L}_r = \left| t_s - \hat{t}_s \right| + \left| t_e - \hat{t}_e \right| \quad (7)$$

in a similar way as previously suggested by parametric 2D lane detection approaches [46, 26]. Finally, the overall loss function is composed of a weighted sum of classification, shape and range loss as

$$\mathcal{L} = \lambda_c \cdot \mathcal{L}_c + \sum_{i=1}^{M} \hat{p}^{(i)} \cdot \left(\lambda_s \cdot \mathcal{L}_s^{(i)} + \lambda_r \cdot \mathcal{L}_r^{(i)} \right), \quad (8)$$

where $\mathcal{L}_s^{(i)}$ and $\mathcal{L}_r^{(i)}$ denote the shape and range loss for the i^{th} line. Despite the fact that the shape loss is computed as a discrete sum of point-wise distances, the parametric formulation reveals important differences to anchor-based

methods, which we want to elaborate here and illustrate in Fig. 3. While anchor-based approaches compute the regression loss at fixed positions independent of the underlying line geometry (green dashed lines), our method flexibly selects the positions, where the loss is evaluated based on the exact location of the ground truth line (green dots in right image). Consequently, for the regression each line instance is represented by the same number of points, i.e. the parametric regression treats each line of arbitrary shape equally. Anchor-based methods, in contrast, pay less attention to sharp curves and short lines only crossing a small subset of anchor positions. Moreover, computing the loss at predefined anchor positions entails systematic errors since the ground truth values $\hat{x}$ and $\hat{z}$ are recovered by interpolation, whereas our continuous formulation allows the direct evaluation of the loss with actual ground truth.

4. Experiments

In this section, we describe our experimental setup, followed by a thorough analysis, in which we investigate our contributions and demonstrate the superior performance of our method compared to two SOTA baselines.

4.1. Experimental setup

Dataset. We evaluate our approach on a synthetic dataset published in the context of Gen-LaneNet [13]. It consists of 10,500 images of highway, urban and rural scenes with 3D ground truth for traffic- and center-lines as well as camera extrinsics, lane visibility information and depth maps. The data is split into three subsets, *(1) Standard* containing simple scenarios from balanced scenes, *(2) Rare Scenes* with more complex road shapes and scenes of *(3) Visual Variations*. The first two share the same training data and only differ in the test set. The third subset yields a training set with well-lighted scenes only, whereas its test set solely holds weakly illuminated scenes.

Evaluation metrics. For the quantitative evaluation, we adhere to the scheme suggested for the utilized dataset [13]. It evaluates the euclidean distance at uniformly distributed points in the range of 0-100 m along the y-direction and counts lanes as matched if the mean distance is below a threshold of 1.5 m. Based on the mean distance, *Average Precision (AP)* and *F-Score* are computed, as well as the mean x- and z-*errors* in *near-* (0-40 m) and *far-range* (40-100 m) to evaluate geometric accuracy.

Implementation details. Similar to [13], we use ERFNet [39] as a backbone with weights pre-trained on the line segmentation task as initialization. The network receives images of size 360×480. We modify the last transposed convolutional layer to obtain an output of 16 feature maps in the same resolution as the image. Regarding the detection head, the number of proposals M should surpass the expected maximum of appearing line instances. In our experiments we found $M = 16$ as an appropriate value. Our best representation uses B-Splines of degree 3 with 15 knots, which makes 18 parameters for $K_{\boldsymbol{\alpha}}$ and $K_{\boldsymbol{\beta}}$. With additional range (2) and line presence (1) parameters the final network output has size 16×39. The vector of normalization constants $\boldsymbol{\eta}$ is set to $[20., 110., 1.]^T$, with η_y restricting the maximum range to 110 m. The regression loss is computed by evaluating the distance for 20 ground truth points equidistantly sampled along the line range. If less points are provided, we interpolate the points. We train our network for 300 epochs on the Standard train set and 200 epochs on the smaller Visual Variations train set and use Adam [21] as optimizer with a learning rate of $1 \cdot 10^{-4}$. The loss weights are set to $\{\lambda_c, \lambda_s, \lambda_r\} = \{1.0, 0.5, 0.06\}$.

Ref.	20 m	First 20 %	**First 40 %**	100 %
F	90.7 %	92.1 %	**92.9 %**	91.2 %
AP	92.5 %	94.1 %	**94.8 %**	93.4 %

Table 1: Comparison of detection scores for different ground truth association references on Rare Scenes test set.

4.2. Ablation study

In this section, we would like to investigate the impact and benefits of our different contributions. For this, we conduct experiments on different ways of ground truth association and compare several continuous representations on the challenging Rare Scenes test set.

Investigation of ground truth association. To analyze the effect of our proposed ground truth association we train models using the same representation with the same hyperparameters and only vary the association reference. We compare a fixed reference point matching commonly used in anchor-based approaches [11, 13] with our association method, which uses the mean distance of a line segment and Hungarian matching to solve the assignment problem. As shown in Table 1 using a fixed point at 20 m results in the lowest F-Score and AP. In contrast, our proposed matching strategy achieves higher scores for all considered ratios using our representation. One main advantage leading to these results is that all lanes regardless of their range and shape are taken into consideration (see discussion in Section 3.3). Furthermore, the Hungarian matching ensures that we obtain an optimal association of the ground truth to our initializations even in challenging situations. However, Table 1 further shows decreasing detection scores when the association ratio is too high. The most likely cause for this is our straight line initialization, which leads to strong overall deviations in curved road situations. More sophisticated initialization strategies should be considered in future that would lead to smaller deviations and, eventually, even

Rep.	d	N	F	x-error		z-error	
				near	far	near	far
Poly.	2	–	88.0	14.0	83.1	2.4	58.1
	3	–	90.4	13.6	75.4	2.6	57.3
	5	–	91.6	9.6	74.8	2.4	58.2
B-Sp.	1	3	81.3	26.3	102.3	3.0	58.1
	3	3	90.8	9.9	69.7	2.4	56.4
	3	5	92.5	8.2	**68.3**	2.2	56.
	1	10	91.5	9.1	71.3	2.2	56.3
	3	10	92.1	8.3	70.8	**1.8**	56.3
	1	15	91.6	9.1	69.1	1.9	**54.9**
	3	**15**	**92.9**	**7.7**	69.9	2.1	56.2

Table 2: Comparison of different representations (Polynomials and B-Splines) on the Rare Scenes test set. d denotes the degree and N the number of knots of the B-Splines. Distance metrics are provided in cm, F-Score in $\%$.

Method	F	AP	x-error		z-error	
			near	far	near	far
3D-L.	86.4	89.3	6.8	47.7	1.5	**20.2**
Gen-L.	88.1	90.1	6.1	49.6	1.2	21.4
Ours	**96.3**	**98.1**	**3.7**	**32.4**	**0.9**	21.3

(a) Standard

3D-L.	72.0	74.6	16.6	85.5	3.9	**52.1**
Gen-L.	78.0	79.0	13.9	90.3	3.0	53.9
Ours	**92.9**	**94.8**	**7.7**	**69.9**	**2.1**	56.2

(b) Rare Scenes

3D-L.	72.5	74.9	11.5	60.1	3.2	**23.0**
Gen-L.	85.3	87.2	7.4	53.8	1.5	23.2
Ours f.b.	**91.3**	**93.1**	**6.9**	**46.8**	**1.3**	24.8

(c) Visual Variations

Table 3: Comparison of 3D-SpLineNet to state-of-the-art methods on all datasets. Ours uses the best representation and ground truth association, i.e. B-Splines with 15 knots of degree 3 and $40\,\%$ mean matching. Distance metrics are provided in cm, F-Score and AP in $\%$.

simpler regression problems also in curved road situations. Finally, we choose the association reference achieving the highest detection scores of the first $40\,\%$ of line range for the following experiments.

Analysis of representations. For the analysis of parametric representations we train several models with different parameterizations for the x- and z-components of the line shape and keep the hyperparameters fixed. More precisely, we consider polynomials, which were previously used in parametric 2D lane detection [46, 26, 29], as well as our main representation based on B-Splines and vary the numbers of knots and degrees.

Table 2 shows an overview of the investigated representations. Obviously, polynomials of degree $2-3$ are not sufficient to model lane geometry, whereas 3^{rd} degree B-Splines show smaller errors even for a low number of 3 knots. Using higher order polynomials indeed yield a better estimate, but also suffer from the dependency between near- and far-range estimations. More specifically, estimating strong deviations, which typically appear in the far-range, induce higher frequencies than necessary to approximate straight parts. This has a negative impact on the near-range approximation, where the road geometry is typically more balanced (see Fig. 1b). Spline models, in contrast, decouple the near- and far-range approximation due to the piecewise formulation. Therefore, splines of lower degrees are already capable to model the entire lane range appropriately, given a sufficient number of knots. We deduce that these circumstances formed the bottleneck in terms of geometrical accuracy of earlier approaches using solely polynomials as parametric representations for 2D lane detection.

Noteworthy is also the influence of the degree on splines. Particularly for compact representations with a low number of knots, using 3^{rd} degree B-Splines enhances significant improvements of the quantitative metrics and provides smooth curves. In contrast, for 1st degree B-Splines, which correspond to discrete poly-lines and hence resemble the anchor-based representation of Gen-LaneNet [13], a higher amount of knots is necessary to provide a sufficient resolution of 3D lane geometry. Table 2 further shows a tendency that with a higher number of knots the benefits in performance become less significant. We deduce that choosing too many knots complicates the learning process, and thus, we do not consider representations of higher capacity.

Following our ablation study, we choose the representation achieving the highest F-Score, i.e. B-Splines of 3^{rd} degree with 15 knots. Still, we want to highlight the comparable performance of lower capacity representations (e.g. 5 knots, 3^{rd} degree) providing a good compromise between number of parameters, degree and geometric accuracy.

4.3. Comparison to state-of-the-art methods

We compare our method to two approaches, which previously achieved state-of-the-art performance, namely 3D-LaneNet [11] and Gen-LaneNet [13]. For the comparison, we used the models optimized for the synthetic dataset provided by the authors of [13].

Table 3 shows a comparison on the three datasets. While we can observe great improvements on all three datasets in terms of F-Score and AP, our approach shows its full potential on Rare Scenes test set with an improvement of $14\,\%$ in F-Score. As discussed in Section 4.2, one reason for the

Method	3D-L.	Gen-L.	Ours
Runtime	41.9 fps	36.3 fps	**74.3 fps**

Table 4: Comparison of runtime in frames per second (fps).

strong detection performance is our proposed association scheme, which results in more correct matches, as shown in Fig. 4a and 4b for the rightmost lane marking. From the lower geometric errors, we further conclude that our parametric representation allows us to better learn line shapes that deviate considerably from the expectations implied by the straight anchor design used by the baselines. This claim is supported by the substantial advantage of estimating x-displacement in the far-range, where anchors match poorly. Fig. 4a and 4d show how our continuous spline representation better captures the straight part followed by a steep curve and Fig. 4b demonstrates its capability to even predict strong height deviations. Our representation also provides smoother transitions than discrete anchor-points as visible in Fig. 4a (top-view) and Fig. 4b (front-view). Besides, Fig. 4c and 4d demonstrate successful handling of challenging occlusions caused by other traffic participants. More qualitative results are provided in the supplementary.

The Visual Variations dataset was utilized in [13] to demonstrate the ability of Gen-LaneNet's two-stage framework to handle domain adaptation and lack of 3D ground truth. Due to the different domains of train and test set with respect to illumination, we follow the setup of Gen-LaneNet and use a fixed backbone (f.b.) pre-trained on the Standard train set. Even though we do not address the domain adaptation topic in this work, our representation dominates in terms of detection performance, while achieving comparable geometric errors compared to Gen-LaneNet. Thus, we see great potential in our representation to tackle the domain adaptation problem for 3D lane estimation in future.

In Table 4, we also provide a runtime evaluation. We ensured the same test conditions for each method providing the same input, expecting the same output and running it on the same device (NVIDIA GeForce Titan X). The results indicate highest processing speed of our method, which does not require expensive interpolation in contrast to the others.

5. Conclusions and future work

We presented 3D-SpLineNet, a method to detect traffic lines in 3D space using a new parametric continuous curve representation. Our sophisticated ground truth association strategy improves detection rate by considering all lines of arbitrary shape in contrast to previous methods. We investigated different representations, among which B-Splines serve as best model to capture even complex line shapes. With this representation, our method achieves state-

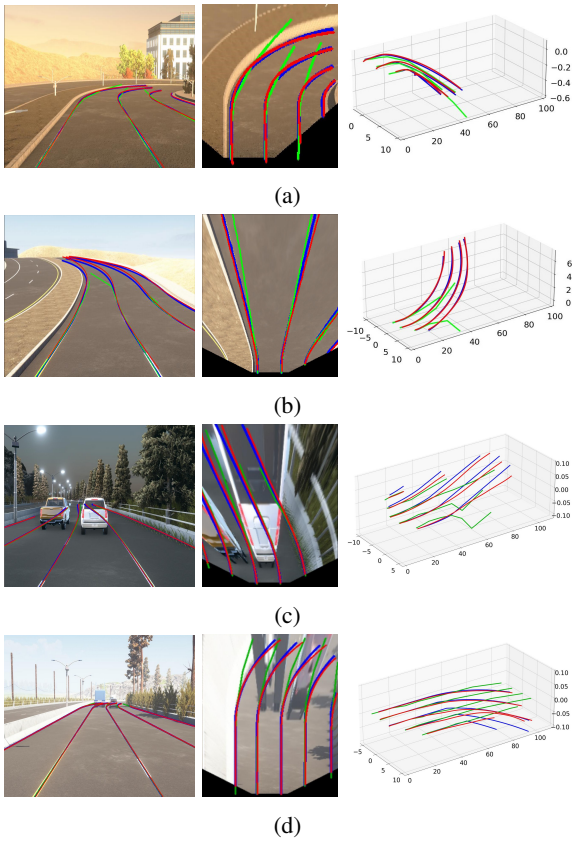


Figure 4: Comparison of *3D-SpLineNet* and *Gen-LaneNet* [13] predictions and *ground truth* on the Rare Scenes test set illustrated in front-view, top-view and 3D space.

of-the-art performance on all test datasets regarding detection score and nearly all geometric errors, where the greatest advantage can be observed on challenging Rare Scenes with an improvement of $14\,\%$ in F-Score. Not requiring costly post-processing, our method also achieves highest processing speed. The qualitative results visually confirm our findings showing significant achievements in the 3D estimation of challenging line shapes, even in cases of occlusions.

As mentioned in the experimental section we expect that more sophisticated initialization strategies would enable lower regression errors. With the first real 3D lane datasets recently getting accessible to the community, we also plan to investigate the performance of our approach in more complex real traffic scenarios soon. In addition, our parametric curve representation provides potential for further improvement since valuable prior knowledge about 3D lane geometry is simply integrable. This is due to the continuous and parametric nature of the line model that simplifies the analytical formulation of geometry enhancing priors such as line parallelism. First experiments show promising results and we plan to elaborate on this subject in future.

References

[1] Mohamed Aly. Real time detection of lane markers in urban streets. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2008.

[2] Aharon Bar-Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications (MVA)*, 25, 2014.

[3] Pierre Coulombeau and Claude Laurgeau. Vehicle yaw, pitch, roll and 3d lane shape recovery by vision. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2002.

[4] Carl de Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[5] Hendrik Deusch, Jürgen Wiest, Stephan Reuter, Magdalena Szczot, Marcus Konrad, and Klaus Dietmayer. A random finite set approach to multiple lane detection. In *Proc. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2012.

[6] E.D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen. The seeing passenger car 'vamors-p'. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 1994.

[7] ED Dickmanns and A Zapp. Guiding land vehicles along roadways by computer vision. In *Congres Automatique*, 1985.

[8] Netalee Efrat, Max Bluvstein, Shaul Oron, Dan Levi, Noa Garnett, and Bat El Shlomo. 3d-lanenet+: Anchor free lane detection using a semi-local representation. *arXiv/2011.01535*, 2020.

[9] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[11] Noa Garnett, Rafi Cohen, Tomer Pe'er, Roee Lahav, and Dan Levi. 3d-lanenet: End-to-end 3d multiple lane detection. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[12] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. EL-GAN: embedding loss driven generative adversarial networks for lane detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[13] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[14] Bei He, Rui Ai, Yang Yan, and Xianpeng Lang. Accurate and robust lane detection based on dual-view convolutional neutral network. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[15] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[16] Albert S. Huang, David Moore, Matthew E. Antone, Edwin Olson, and Seth J. Teller. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2-3):103–122, 2009.

[17] Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo. Multi-lane detection in urban driving environments using conditional random fields. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[18] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, Fernando A. Mujica, Adam Coates, and Andrew Y. Ng. An empirical evaluation of deep learning on highway driving. *arXiv/1504.01716*, 2015.

[19] Heechul Jung, Junggon Min, and Junmo Kim. An efficient lane detection algorithm for lane departure detection. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[20] Surender K Kenue. Lanelok: Detection of lane boundaries and vehicle tracking using image-processing techniques-part i: Hough-transform, region-tracing and correlation algorithms. In *Mobile Robots*. SPIE, 1990.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015.

[22] YeongMin Ko, Jiwon Jun, Donghwuy Ko, and Moongu Jeon. Key points estimation and point instance segmentation approach for lane detection. *arXiv/2002.06604*, 2020.

[23] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[24] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017.

[25] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Trans. on Intelligent Transportation Systems (T-ITS)*, 21(1):248–258, 2020.

[26] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[28] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[29] Pingping Lu, Chen Cui, Shaobing Xu, Huei Peng, and Fan Wang. SUPER: A novel lane detection system. *IEEE Trans. on Intelligent Vehicles (T-IV)*, 6(3):583–593, 2021.

[30] Hanspeter Mallot, Heinrich Bülthoff, J.J. Little, and S Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64:177–85, 1991.

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

[32] Sergiu Nedevschi, Rolf Schmidt, Thorsten Graf, Radu Danescu, Dan Frentiu, Tiberiu Marita, Florin Oniga, and Ciprian Pocol. 3d lane detection system based on stereovision. In *IEEE Trans. on Intelligent Transportation Systems (T-ITS)*, 2004.

[33] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[34] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial CNN for traffic scene understanding. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2018.

[35] Fabio Pizzati, Marco Allodi, Alejandro Barrera, and Fernando García. Lane detection and classification using cascaded cnns. In *Proc. of the International Conf. on Computer Aided Systems Theory (EUROCAST)*, 2019.

[36] D. Pomerleau. Ralph: rapidly adapting lateral position handler. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 1995.

[37] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeuIPS)*, 2015.

[39] Eduardo Romera, Jose M. Alvarez, Luis Miguel Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. on Intelligent Transportation Systems (T-ITS)*, 19(1):263–272, 2018.

[40] I. J. Schoenberg. *Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions*, pages 3–57. Birkhäuser Boston, Boston, MA, 1988.

[41] Jongin Son, Hunjae Yoo, Sanghoon Kim, and Kwanghoon Sohn. Real-time illumination invariant lane detection for lane departure warning system. *Expert Systems With Applications (ESA)*, 42(4):1816–1824, 2015.

[42] Jinming Su, Chao Chen, Ke Zhang, Junfeng Luo, Xiaoming Wei, and Xiaolin Wei. Structure guided lane detection. In *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI)*, 2021.

[43] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[44] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, and Keqiang Li. A novel curve lane detection based on improved river flow and RANSA. In *Proc. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2014.

[45] Jigang Tang, Songbin Li, and Peng Liu. A review of lane detection methods based on deep learning. *Pattern Recognition*, 2021.

[46] Lucas Tabelini Torres, Rodrigo Ferreira Berriel, Thiago M. Paixão, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *Proc. of the International Conf. on Pattern Recognition (ICPR)*, 2020.

[47] Andreas Wedel, Hernán Badino, Clemens Rabe, Heidi Loose, Uwe Franke, and Daniel Cremers. B-spline modeling of road surfaces with an application to free-space estimation. *IEEE Trans. on Intelligent Transportation Systems (T-ITS)*, 10(4):572–583, 2009.

[48] Lu Xiong, Zhenwen Deng, Peizhi Zhang, and Zhiqiang Fu. A 3d estimation of structural road surface based on lane-line information. *IFAC Conf. on Engine and Powertrain Control, Simulation and Modeling (E-COSM)*, 2018.

[49] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Trans. on Vehicular Technology (VTC)*, 69(1):41–54, 2020.