The Visual Computer (2023) 39:581–595
https://doi.org/10.1007/s00371-021-02358-1

ORIGINAL ARTICLE

Improving lane detection with adaptive homography prediction

Yiman Chen[1] · Zhiyu Xiang[2] · Wentao Du[1]

Accepted: 10 November 2021 / Published online: 10 January 2022

Abstract
Lane marks regulate the routes and define the prior drivable areas for a vehicle. Robust detection of lanes plays a vital role in intelligent vehicle navigation. Lane detection algorithms are usually composed of two steps: lane candidate generation and lane curve fitting. The latter is not only used for fitting lane mark candidates with concise curve forms but also for removal of the outliers produced in the former step. Therefore, lane curve fitting is crucial for lane detection. In this step, a common way is carrying out the curve fitting on the bird's-eye view (BEV), which can mitigate the distortion caused by the perspective projection and improve the fitting results. However, due to the sloping road surfaces in real scenarios, the relative pose between the camera and the ground can change frequently, where using a fixed pre-calibrated projection matrix could bring extra errors in curve fitting. In this paper, we propose a homography prediction network named HP-Net for robust lane mark fitting under various sloping roads. The network can adaptively predict the homographic projection matrix for each input image, producing a suitable BEV for lane fitting. Considering the parallel nature of multiple lanes, the HP-Net could skillfully be trained by reusing the lane labels originally for the task of lane mark segmentation, without introducing any extra manpower. Our method has been verified on a large dataset CULane and another dataset acquired by ourselves. Experiment results show that the proposed model can effectively improve the robustness and accuracy of lane detection.

Keywords Deep learning · Lane detection · Curve fitting · Homography prediction

1 Introduction

Lane detection is an important task in the assistant driving system (ADS) for intelligent vehicles. Accurate and robust perception of lanes is crucial for reliable vehicle navigation. However, lane detection in practice still remains a challenge due to the diverse appearance of lane marks and complicated driving scenarios. Lane marks have a long and thin shape with a high degree of freedom. The pattern of the lane marks can be yellow or white, solid or dashed, straight or curving, merging or detaching. Except for the internal nature of lane mark, the external complex environment should also be considered. Bad weather conditions, poor illumination conditions, and other unexpected factors would all affect the appearance of lane marks, posing a great difficulty for the detection task.

A variety of researches on lane detection have been proposed to handle these problems. Early conventional solutions extract low-level features of lane marks like color or edges[1, 2]. These hand-crafted cues can be combined with Hough Transform or filters[3, 4] to generate lane segment candidates. Then, post-processing techniques are used to eliminate the detection errors and connect these lane segments together to form the final results. These traditional methods depend largely on hand-crafted features, which can only work in limited scenarios with strict constraints.

With neural networks, works on robust learnable features have shown great potential to handle tasks of computer vision[5–7]. Recent advances in lane detection can also be attributed to the development of convolutional neural networks (CNN). Kim et al.[8] use CNN to extract lane candidate regions, followed by random sample consensus (RANSAC) and line fitting. Jun Li et al.[9] adopt CNN and RNN to detect the lane boundaries with line clustering. VPGNet[10] is proposed to jointly detect lane marks and other road markings, where a series of post-processing (e.g., point sampling,

✉ Zhiyu Xiang
xiangzy@zju.edu.cn

Yiman Chen
chenyiman@zju.edu.cn

[1] College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China

[2] Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking, Zhejiang University, Hangzhou, China

clustering, lane regression) was employed. Spatial CNN (SCNN)11 is designed to capture the spatial relationships of thin- and long-shaped lane marks, where the positions with the highest probability response were searched and connected by cubic splines to generate the final results.

From what is discussed above, it can be discovered that most of the methods for lane detection have a two-step pipeline, i.e., lane candidate generation and lane curve fitting. Having estimated the candidate position or probability map, it is necessary to describe them by a parametric expression through a fitting model. Curve fitting is crucial because it is not only for the simpler description of lane mark curve but also for achieving better detection accuracy through filtering out the outliers. Most of the CNN-based methods only predict candidates of the lane mark, the step of fitting them into lane curves still relies on non-learning methods. Cubic polynomials, splines and clothoids are some of the most popular fitting models12–14. However, due to perspective distortion, fitting lane curves directly into the image space could be an inferior choice. A better alternative is converting the original image into BEV using an inverse perspective projection and then performing curve fitting there. Typically, camera calibration and calculation of the homographic transformation matrix are both implemented before the vehicle starts moving. However, a fixed homography could not well remove the perspective distortion effect when the relative pose between vehicle and ground plane varies (e.g., by hilly ground or shaking of the camera), which will lead to inaccurate lane fitting. There are also situations where the camera cannot be pre-calibrated in advance.

In this paper, HP-Net is proposed to predict the crucial parameters of the adaptive homographic projection matrix for each input image. With a more accurate homographic prediction matrix, lane marks can be fitted in a more realistic top-view space where the curves are free of perspective distortions. Utilizing the parallelism nature of lanes, the network can be trained without providing any extra annotations except for the original lane labels. Finally, building upon SCNN11, an improved lane detection algorithm is realized. Experiments are carried out on a large-scale CULane dataset11 and our own dataset. The evaluation results show that the proposed method can effectively improve the accuracy of lane detection. The contributions of our work can be summarized as follows:

We integrate the perspective projection geometry into a deep learning framework. A neural network that can learn to predict parameters of the homographic transformation matrix between the input image and the ground is proposed. It is adaptive to the pose changes between the camera and the road plane.

An annotation-sharing method is proposed to train the proposed network. Utilizing the parallel nature of multiple lanes, the lane annotations for the detection task could be reused in this homography prediction task. No extra manual annotation is required.

Combined with our adaptive homography prediction with a lane instance segmentation network, a novel lane detection pipeline is constructed. Our method is tested on a large-scale public dataset CULane and our own dataset, achieving the state-of-the-art performance.

2 Related work

Traditional methods for lane detection often involve four procedures: image pre-processing, hand-crafted feature extraction, curve fitting, and post-processing15. Choosing a region of interest (ROI)16 helps to wipe off invalid non-lane areas and reduces computation. Hand-crafted feature like color is utilized for lane mark retrieval17. Filtering methods and Hough Transform are further employed to extract line features18. These model-driven approaches require sophisticated pre-processing with strong geometric assumptions to determine the positions of lane marks. They are sensitive to noises, which mortifies their performances in complex environments.

The emergence of deep neural networks and large-scale datasets provide a more feasible solution to lane detection. Recent progress for lane detection mainly focus on pixel-wise segmentation-based methods 11, 19, 20. SCNN11 utilizes slice-wise convolutions in a segmentation module and aggregates information from different dimensions through processing slice features. LaneNet19 adopts a shared encoder for feature extraction, followed by two decoders: one for binary lane segmentation and the other for pixel embedding. Hsu et al.20 employ the pairwise relationship between pixels to train the network for image pixel clustering, which shows the effectiveness in both lane detection and generic instance segmentation. Other deep learning-based approaches solve the problem of lane detection from different aspects[21–23]. StripNet[21] treats lane detection as a local boundary regression problem, predicting the boundary of target strip sequences after ROI alignment. Line-CNN[22] puts forward a novel line proposal unit (LPU), which uses line proposals as references to predict horizontal coordinate offsets for lanes. Qin et al.[23] propose a lightweight lane detection scheme using row-based selecting with global image features.

Although hand-crafted feature extraction has been largely overtaken by neural networks, post-processing for lane detection has made little progress during recent years. Due to the slender shape of lane marks and the imbalance with background, segmentation results are often unsmooth with noises. To eliminate the errors and integrate the detection results with other perception tasks, curve fitting is necessary. Frequently-used fitting models include straight lines,

polynomial curves, splines, clothoids, and snakes13, 14, 24, 25. Least mean square(LMS), random sample consensus(RANSAC), Hough transform and Kalman filter, etc.3, 4, 12, 25 are the most commonly adopted fitting algorithms. Instead of fitting features, some end-to-end methods directly predicted parameters of lanes26, 27. PolyLaneNet 26 regards lane detection as a polynomial regression problem. Liu et al.27 formulate the lane shape model based on road structures and camera pose, using a transformer to capture slender structures and a richer context. Compared to these works that directly output fitted parameters on image space, we prefer implementing the fitting on an adaptive projected ground plane.

Lane curve fitting in image space often suffers from accuracy degradation because of the inconsistent scales induced by perspective projection. An intuitive solution to this problem is converting the original image to BEV by using the inverse perspective mapping (IPM). It is frequently adopted as a part of pre-processing both in conventional and neural network schemes13, 28. Tom Bruls et al.29 propose an adversarial learning approach to generate an improved IPM image using the Spatial Transformer Network (STN)30. They adopted visual odometry to obtain ground-truth BEV images for supervision and train the network with a GAN loss. In contrast to these IPM-based works, our work leverages fundamental computer vision theories and integrates prior geometric knowledge into a deep learning framework, which can effectively predict an adaptive homographic projection for lane fitting without BEV ground truth.

3 Methods

An overview of our entire lane detection framework is illustrated in Fig. 1. Given an input image, SCNN [11] is adopted to generate lane probability maps and predict the existence of lane marks. Pixels whose probability responses are the highest in the local area along each row are preserved for fitting. These candidate lane pixels are projected onto a homographic matrix produced by HP-Net. Instead of directly fitting in image space, we perform curve fitting in BEV. Finally, the fitted lanes are projected back to the original image space to obtain the final detection results. Thanks to the predicted homographic projection, the lane marks could be fitted in a more simple polynomial form and be less affected by the varied slope of the road plane.

3.1 Network model for homography prediction

Having estimated which pixels belong to the lane mark, we are supposed to fit these pixels into a parametric curve with outliers removed. The lane pixels are first projected into a BEV representation, in which the lanes have a parallel-like shape and their curvature can be fitted with low-order polynomials. If a fixed transformation matrix is employed, the projection becomes less accurate when sloping ground-planes or camera vibrations are encountered. To remedy this situation, we train a network to output certain crucial parameters in the perspective transformation.

A full projection model describes the mapping from world to pixel coordinates. For this nonlinear projection, more unknown model parameters mean a higher risk of unstable

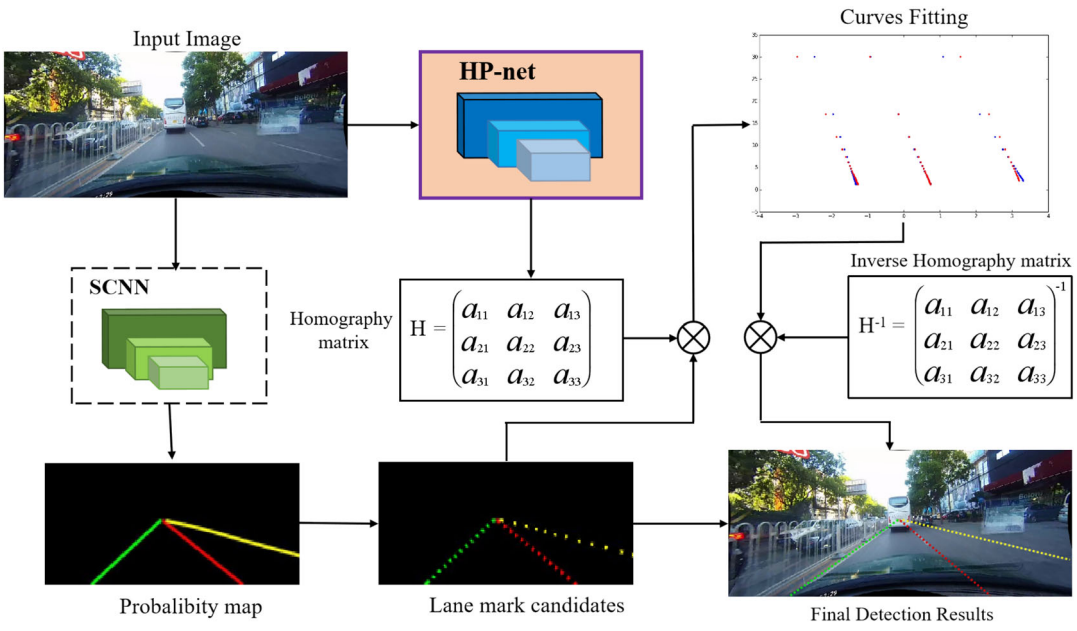


Fig.1 Overview of the framework

output. Fully constructing a 3×3 homographic matrix needs 8 dependent components. However, treating each of them as an independent parameter is not a good idea since they are actually correlated. Therefore, we set up a homographic model from the fundamental projection principle and try to reduce the number of outputs to the least degrees of freedom in $\boldsymbol{H}$. Depending on whether the camera is pre-calibrated, different outputs of the model are designed.

3.1.1 Homography model for a calibrated camera

Setting a world coordinate system $\boldsymbol{X_G} = (\mathrm{x_G}, \mathrm{y_G}, \mathrm{z_G})$ attached to the ground, a rigid body transformation brings a point from the ground to the camera coordinate system by a rotation matrix and a translation vector. Let α, β, θ represent the roll, pitch and yaw angles, respectively, with the translation vector $\boldsymbol{T} = [\mathrm{t_1}, \mathrm{t_2}, \mathrm{t_3}]$ and the assumption $\boldsymbol{Z_G} = 0$ for the ground plane, the mapping from the ground frame to the pixel frame is given by:

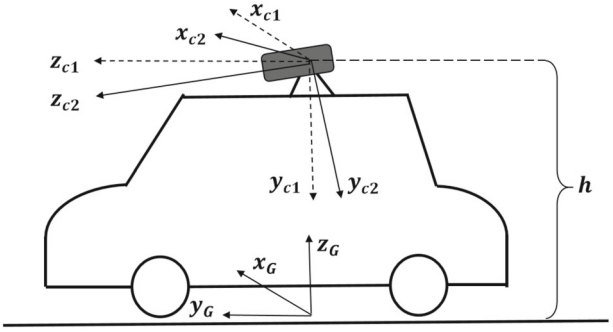


Fig.2 Transformation from the word to the camera

As shown in Fig. 2, setting the original point of the world frame $\boldsymbol{X_G}$ right on the ground under the camera, the relationship between $\boldsymbol{X_G}$ and the virtual horizontal camera frame $\boldsymbol{X_{C1}}$ is:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta\cos\theta & \cos\beta\sin\theta & t_1 \\ -\cos\alpha\sin\theta + \sin\alpha\sin\beta\cos\theta & \cos\alpha\cos\theta + \sin\alpha\sin\beta\sin\theta & t_2 \\ \sin\alpha\sin\theta + \cos\alpha\sin\beta\cos\theta & -\sin\alpha\cos\theta + \cos\alpha\sin\beta\sin\theta & t_3 \end{bmatrix} \begin{bmatrix} x_G \\ y_G \\ 1 \end{bmatrix} \tag{1}$$

where f_x, f_y and $(\mathrm{u_0}, \mathrm{v_0})$ separately represent the focal length along x_c and y_c direction and the principle point coordinates in the image plane.

The homographic projection matrix $\boldsymbol{H}$ projecting the image plane to the ground can be expressed as:

$$H = \left(\begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta\cos\theta & \cos\beta\sin\theta & t_1 \\ -\cos\alpha\sin\theta + \sin\alpha\sin\beta\cos\theta & \cos\alpha\cos\theta + \sin\alpha\sin\beta\sin\theta & t_2 \\ \sin\alpha\sin\theta + \cos\alpha\sin\beta\cos\theta & -\sin\alpha\cos\theta + \cos\alpha\sin\beta\sin\theta & t_3 \end{bmatrix} \right)^{-1} \tag{2}$$

Given a calibrated camera, the intrinsic matrix and the initial position of the camera with respect to the ground frame can be well known. Only the change in relative rotations needs to be considered during driving. Therefore, the network is trained to predict three rotation angles: the roll angle α, the pitch angle β and the yaw angle θ. Once they are predicted, $\boldsymbol{H}$ can be reconstructed according to Eq. 2.

3.1.2 Homography model for an uncalibrated camera

For open-source datasets where the parameters of the camera are not available, more unknowns are supposed to be considered. To predict a stable network output, it is necessary to reduce the unknowns with some reasonable assumptions. Firstly, among all three rotation angles, the depression angle between the camera and the ground plane is the most influential, and the other two can be assumed to be zero.

$$R_1 \begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix} + T_1 = \begin{bmatrix} x_{c1} \\ y_{c1} \\ z_{c1} \end{bmatrix} \tag{3}$$

With $R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, $T_1 = \begin{bmatrix} 0 \\ h \\ 0 \end{bmatrix}$, where h is the height of the camera to the ground.

Defining the rotation angle about the x-axis of the virtual horizontal camera frame $\boldsymbol{X_{C1}}$ as pitch angle θ, the transformation between the new rotated camera frame $\boldsymbol{X_{C2}}$ and $\boldsymbol{X_G}$ can be expressed with homogeneous coordinates as:

$$\begin{bmatrix} x_{c2} \\ y_{c2} \\ z_{c2} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin\theta & -\cos\theta & h\cos\theta \\ 0 & \cos\theta & \sin\theta & -h\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_G \\ y_G \\ z_G \\ 1 \end{bmatrix} \tag{4}$$

Next, assuming $f_x = f_y = f$ and $\mathbf{Z_G} = 0$, the mapping from world coordinates to pixel coordinates becomes:

$$z_{c2}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\theta & h\cos\theta \\ 0 & \cos\theta & -h\sin\theta \end{bmatrix}\begin{bmatrix} x_G \\ y_G \\ 1 \end{bmatrix} \tag{5}$$

As a result, for an uncalibrated camera, the homographic projection matrix $\boldsymbol{H}$ is:

$$\begin{aligned} H &= \left(\begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\theta & h\cos\theta \\ 0 & \cos\theta & -h\sin\theta \end{bmatrix}\right)^{-1} \\ &= \left(\begin{bmatrix} f & u_0\cos\theta & -u_0 h\sin\theta \\ 0 & f\sin\theta + v_0\cos\theta & fh\cos\theta - v_0 h\sin\theta \\ 0 & \cos\theta & -h\sin\theta \end{bmatrix}\right)^{-1} \end{aligned} \tag{6}$$

For simplicity, the coordinates of the principle points are set as $(u_0, v_0) = (W/2, H/2)$, where W and H are the known width and height of the image in the pixel unit. In public large-scale datasets like CULane, vehicles mounted with different cameras are used during data collection, therefore the camera height h and focal length f are unknown and changeable. Finally, in the case of an uncalibrated camera, our network is trained to predict three parameters for homography: f, θ and h.

It should be noted that our aim is to predict some crucial parameters for reconstructing the homography matrix rather than accomplishing an accurate camera calibration. As long as the reconstructed homographic matrix is able to map the imaging lanes to parallel ones on the ground plane, the predictions can be acceptable.

3.1.3 Network architecture

The network architecture of HP-Net is illustrated in Fig. 3. Three convolution blocks are designed to extract the features of the input image. Each block consists of two 3 × 3 convolution layers and one 2 × 2 max-pooling layers to decrease the dimension. Batch-normalization and ReLUs are used for each convolution layer. At the end of the network, we adopt a global average pooling layer and a linear output layer. As described in the previous section, HP-Net predicts three different parameters of the homographic matrix, depending on whether the camera is calibrated.

3.2 Loss functions and annotation-shared training

HP-Net takes the entire image as input and is trained with a loss function that is tailored to the lane fitting problem. Training of the network is challenging since it is extremely difficult to obtain the required ground truth for the homographic matrix. Inspired by the self-supervising training, we develop a training approach that only needs the annotations of lane marks, which is originally just for the segmentation task.

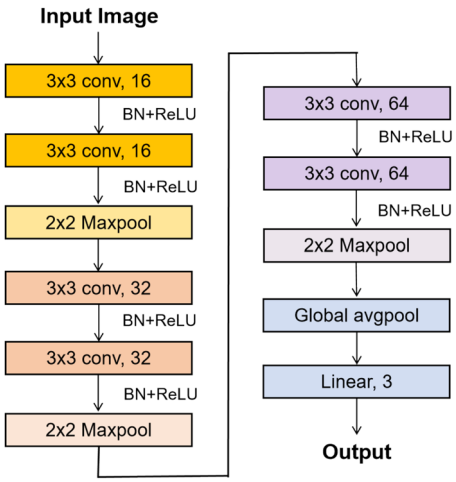


Fig. 3 The network architecture

As shown in Fig. 4, HP-Net takes the RGB image as input and predicts the parameters to generate the homography matrix. Then, the lane pixels are projected onto the obtained BEV space and fitted by a group of parallel lines. Finally, we re-project the fitted lanes back to the original image space via the inverse homography matrix and compute the loss by comparing the results with the ground truth of lanes. Through this way, the model can be trained effectively. We explain this process in detail in the following section.

3.2.1 Curve fitting

Ground-truth lane points are defined as P where each point is denoted with $p_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T \in P$. With the homographic matrix $\boldsymbol{H}$, the projected pixel $p'_i = \begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix}^T = Hp_i \in P'$ can be obtained. The least-squares algorithm is then used to fit a group of polynomials through the transformed pixels P'. The polynomial curves are sampled at different y-positions y'_i to get the fitted x-position x'^*_i with $x'^*_i = f(y'_i)$. Then with the fitted points P'^* and each point $p'^*_i = \begin{bmatrix} x'^*_i & y'_i & 1 \end{bmatrix}^T \in P'^*_i$, we re-project them back to the original image space via the inverse transformation matrix to get: $p^*_i = H^{-1}p'^*_i$ where $p^*_i = \begin{bmatrix} x^*_i & y_i & 1 \end{bmatrix}^T$. Note that the y-positions of lane points remain the same while the x-positions are changed after curve fitting. The above process is illustrated in Fig. 5.

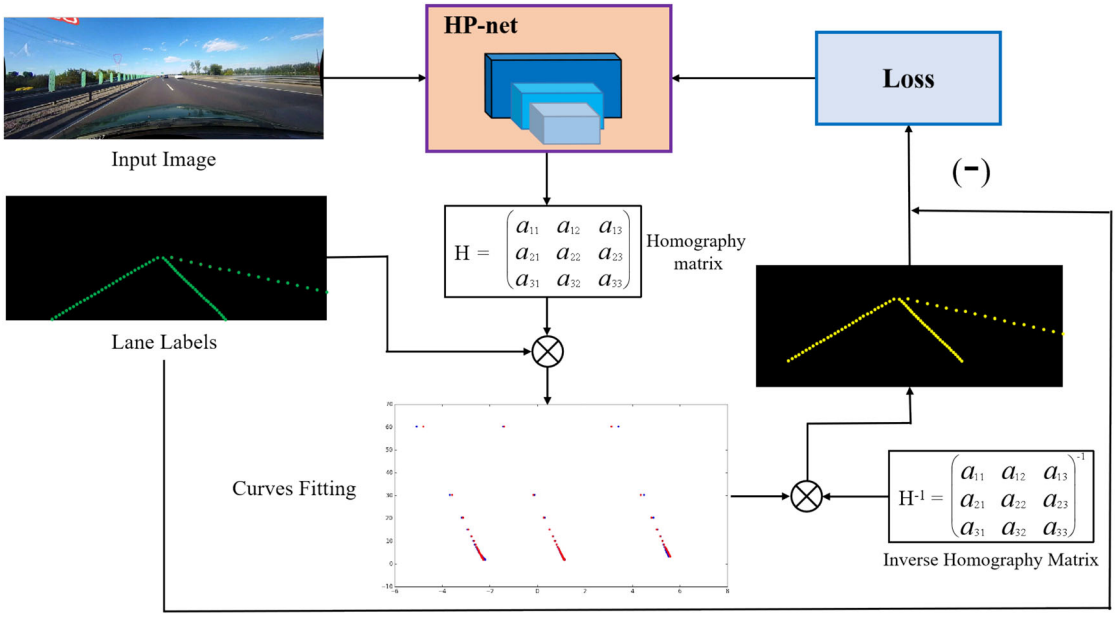


Fig.4 Scheme of network training

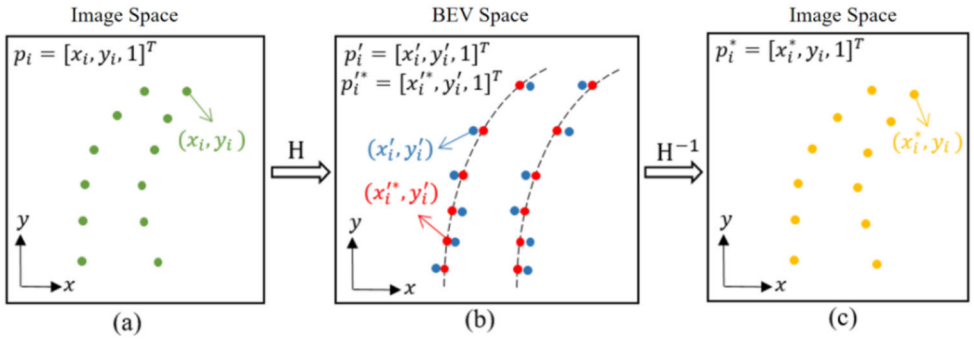


Fig.5 Illustration of curve fitting. **a** Ground-truth lane points (*green*) in the original image are projected into BEV space by the predicted transformation matrix H. **b** A group of parallel curves are jointly fitted upon the transformed points (*blue*) to obtain the fitted points (*red*). **c** The fitted points are projected back to the original image space (*marked by yellow points*) and compared with the lane labels to produce the training loss

3.2.2 Loss function

In order to train HP-Net, the loss function has been properly designed. One innovation for our network is that it could be trained by sharing the lane mark annotations with the segmentation task. Thus, no extra annotation work is required.

In the projected ground plane, we jointly fit all the lane curves on the image with the same polynomial parameters, with the assumption that these lanes are parallel to each other. As described earlier, the ground-truth lane points $p_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T \in P$ in each image are firstly projected to: $p'_i = \begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix}^T = Hp_i \in P'$. Then the lane mark fitting with polynomials can be applied. Taking the 3rd order polynomials with the form $f(y') = ay'^3 + by'^2 + cy' + d$ as an example, the joint curve fitting process is described as follows:.

Given k lane and N lane points on an image, the coefficients w of 3rd polynomials can be computed by a closed-form least square solution as:

$$w = \left(Y^T Y\right)^{-1} Y^T x' \tag{7}$$

where $w = \begin{bmatrix} a & b & c & d_1 & d_2 & \cdots & d_k \end{bmatrix}^T$, $x' = \begin{bmatrix} x'_1 & x'_2 & \cdots & x'_N \end{bmatrix}$, and

$$Y = \begin{bmatrix} y_1'^3 & y_1'^2 & y'_1 & 1 & 0 & \cdots & 0 \\ y_2'^3 & y_2'^2 & y'_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ y_N'^3 & y_N'^2 & y'_N & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{8}$$

In the expressions of $\boldsymbol{Y}$, row vector [1 0 0 0…0] is appended to the rows whose y-positions (or their power) are on the first lane, and [0 1 0 0…0], [0 0 1 0…0], …, [0 0 0 0…1] are for the 2nd,3rd …, and kth lane, respectively. With Eq. 8, we use the assumption that the lane curves are parallel and thus can have the same coefficients in the polynomial except for the constant terms. In other words, once the curves can be jointly fitted by a polynomial with little error, we conclude that the predicted homographic matrix has successfully projected the image to the ground plane.

After fitting, we can get the fitted prediction $x_i'^*$ for each y_i' location with the resulting w:

$$x'^* = Y \times w \tag{9}$$

To utilize the lane annotations in image space as the supervisory signal, the fitted points in the ground plane are then projected back to the original image by the inverse homographic matrix with $p_i^* = H^{-1} p_i'^*$, where $p_i'^* = \begin{bmatrix} x_i'^* & y_i' & 1 \end{bmatrix}^T$ and $p_i^* = \begin{bmatrix} x_i^* & y_i & 1 \end{bmatrix}^T$ are the lane points before and after the projection. Finally, keeping the y-positions unchanged, the differences between x^* and the ground truth x could be calculated to construct L_2 loss:

$$Loss = \frac{1}{N} \sum_{i=1}^{N} \left(x_i^* - x_i \right)^2 \tag{10}$$

In this way, HP-Net can be trained by sharing the supervisory annotations originally employed for the lane detection tasks. The comparison among the use of L_1 or L_2 loss as well as the different polynomial fitting models will be presented in Sect. 4.

Note that in the training stage, we assume the lanes are parallel. This is reasonable and practical for most of the training images in datasets. For some special scenes like cut-in or spread-out lanes, we find the training can still proceed well. When the network works in the inference model, fitting with separate polynomials for each lane could be carried out to handle those special merging or splitting lanes.

4 Experiments

4.1 Dataset and experimental setup

First, we adopted CULane11, which is a challenging large-scale dataset for lane detection. The images with a resolution of 1640 × 590 are captured by different uncalibrated cameras mounted on vehicles. In each image, the ego-lane as well as its left and right lane are annotated as ground truth. Both SCNN and HP-Net are trained separately. The original training set of CULane is used to train SCNN. As for HP-Net, we selected 2340 images for training, 300 for validation. The original images are down-sampled to 128 × 64 to accelerate the training and inference process.

To evaluate the model for the calibrated camera, we built our own dataset to further verify the proposed method. A calibrated camera is mounted on the top of a car to collect road image data. The images are acquired under a wide range of scenarios including sloping ground, shadow, glare, tunnel, bridge, and occlusion. The original images are down-sampled into 512 × 384 before storing. Some examples with annotated ground truths are shown in Fig. 6. Here, SCNN is pre-trained by images from CULane and fine-tuned on our own dataset. We divided our own dataset into three subsets: 2187 for training, 274 for validation, and 1093 for testing.

Our model is implemented on Tensorflow 31 framework. The network is trained using Adam optimizer with a learning rate of 1e-8 and batch size of 5. During testing, SCNN is run first to output probability maps of lane curves. Then we keep those lanes whose confidence is larger than 0.5. For fixed intervals along each lane, the positions with the local highest probability are sampled as lane candidate points for curve fitting. Using the homographic matrix H predicted by HP-Net, these candidate lane points are fitted to the projected ground plane. In this step, each curve is fitted separately to handle the non-parallel lane situation.

4.2 Ablation for the loss function

During training we used polynomial functions for projected lane candidate fitting, where 2nd or 3rd polynomials could be employed. Then the fitted points are projected back to the original image space to compute the L_1 or L_2 loss with the ground truth. To select the best form of the entire loss function, we evaluated these different training settings under F1 metric. The correct lane predictions are regarded as those whose intersection-over-union (IoU) with GT lanes is higher than a threshold. Here, we consider IoU = 0.7 as a threshold for the strict metric of true positives (TPs). Then F1-measure $= 2 \times \frac{Precision \times Recall}{Precision + Recall}$ is adopted as the final evaluation indicator, where Precision $= \frac{Tp}{TP+FP}$ and Recall $= \frac{Tp}{TP+FN}$. The evaluation results in Table 1 show that using the 3rd polynomial (poly3) and L_2 loss works the best. Therefore, we choose poly3 and L_2 loss as the default training settings for later experiments.

4.3 Evaluation

4.3.1 Comparisons with existing methods

A variety of learning-based methods for lane detection are proposed, including ENet-SAD32, SCNN11, Ultra-Fast23, ERFNet-E2E33, PINet34, CurveLanes35, LaneATT36,

Fig.6 Sample images with lane annotations in our own dataset

Table 1 F1 values on different training settings("%" is omitted). "poly2" or "poly3" in the first column denotes the polynomials used in L_1 or L_2 loss (second column) during training. Fit_* denotes the fitting function used in producing the F1 metric in the final output lane points, i.e., 2nd, 3rd order polynomials or cubic splines, respectively

Training Setting	Fit_2nd	Fit_3rd	Fit_cub	
F1(poly2)	L_1	68.81	76.06	77.61
	L_2	**70.48**	**76.32**	**77.73**
F1(poly3)	L_1	69.16	76.32	77.58
	L_2	**70.77**	**76.70**	**77.87**

SGNet37, FOLOLane38, CondLaneNet39. The results of our approach and state-of-the-art methods on CULane are shown in Table 2. For a fair comparison, the IOU threshold of the F1 measure is set at 0.5 and the fitting model adopts a cubic spline. Except for sub-categories of *hlight* and *noline*, our method achieves the best performance in the F1 measure, showing great robustness to different scenarios. It is worth noting that for some confusing cases, such as crowd and arrow, our method has obvious advantages with 2.91% and 3.58% improvements relative to the second-best one.

4.3.2 Ablation study on CULane

Absolute mean error (AME) metric is used for directly evaluating the accuracy of curve fitting. Given the same y-coordinate, the x-coordinates of the points in the fitted lane curves are compared to the ground truth. The absolute mean error (AME) of x-coordinates is calculated as:

$$AME = \frac{1}{N}\sum_{i=1}^{N}|\Delta x_i| = \frac{1}{N}\sum_{i=1}^{N}\left|x_{fit} - x_{gt}\right| \tag{11}$$

where N is the total number of the sampling points in the image, x_{fit} and x_{gt} are the fitted and ground truth x-coordinates, respectively.

Table 2 Comparisons with state-of-the-art methods on the CULane test set. F1-measure with an IoU threshold of 0.5 is used to evaluate the results of 8 sub-categories and the total

Methods	Normal	Crowd	Hlight	Shadow	Noline	Arrow	Curve	Night	Total
ENet-SAD [32]	90.10	68.80	60.20	65.90	41.60	84.00	65.70	66.00	70.80
SCNN [11]	90.60	69.70	58.50	66.90	43.40	84.10	64.40	66.10	71.60
Ultra-Fast [23]	90.70	70.20	59.50	69.30	44.40	85.70	69.50	66.70	72.30
ERFNet-E2E [33]	91.00	73.10	64.50	74.10	46.60	85.80	71.90	67.90	74.00
PINet [34]	90.30	72.30	66.30	68.40	49.80	83.70	65.60	67.70	74.40
CurveLanes [35]	90.70	72.30	67.70	70.10	49.40	85.80	68.40	68.90	74.80
LaneATT [36]	91.74	76.16	69.47	76.31	50.46	86.29	64.05	70.81	77.02
SGNet [37]	92.07	75.41	67.75	74.31	50.90	87.97	69.65	72.69	77.27
FOLOLane [38]	92.70	77.80	**75.20**	79.30	52.10	89.00	69.40	74.50	78.80
CondLaneNet [39]	93.47	77.44	70.93	80.91	**54.13**	90.16	75.21	74.80	79.48
Ours	**94.51**	**80.71**	75.15	**81.81**	51.11	**93.74**	**76.69**	**74.81**	**79.74**

Bold data indicates that the corresponding results are better

Table 3 AME Errors of different fitting models in image and projected space. The units are in pixels

AME	Normal	Crowd	Hlight	Shadow	Noline	ARROW	curve	Night	Total
Img_2nd	14.964	17.240	29.054	29.846	41.712	14.491	33.681	20.890	19.584
Proj_2nd	**13.646**	**16.109**	**26.313**	**25.429**	**41.117**	**12.250**	**32.078**	**19.421**	**18.082**
Img_3rd	16.317	18.747	30.778	30.797	43.536	16.347	35.433	21.644	20.804
Proj_3rd	**14.235**	**16.945**	**28.550**	**28.385**	**41.365**	**13.353**	**32.882**	**20.214**	**18.967**
Img_cub	18.036	20.447	**31.546**	**32.504**	45.006	**17.932**	35.861	22.515	22.203
Proj_cub	**17.988**	**20.140**	32.145	32.548	**44.829**	18.020	**34.201**	**22.409**	**22.116**

Bold data indicates that the corresponding results are better

Table 4 F1-measure values of different fitting models in image and projected space

F1	Normal	Crowd	Hlight	Shadow	Noline	Arrow	Curve	Night	Total
Img_2nd	69.86	52.32	47.76	37.47	27.87	65.75	42.85	45.44	51.76
Proj_2nd	**76.75**	**59.33**	**49.95**	**45.92**	**29.75**	**72.84**	**48.51**	**49.73**	**57.32**
Img_3rd	69.73	52.52	47.98	37.47	27.80	67.04	44.01	45.14	51.70
Proj_3rd	**76.18**	**59.05**	**48.64**	**46.73**	**29.10**	**72.10**	**50.11**	**49.06**	**56.85**
Img_cub	55.60	39.76	38.60	30.45	23.85	51.20	35.88	35.64	40.79
Proj_cub	**70.45**	**53.32**	**42.31**	**40.81**	**25.52**	**63.26**	**41.98**	**42.68**	**50.98**

Table 5 F1-measure values of the detection by curve fitting with different projections. The suffix "uncal" and "cal" denotes the uncalibrated and calibrated camera

IOU_tr = 0.7	TP	FP	FN	Precision	Recall	F1
Img_2nd	2339	1091	1073	0.68192	0.68552	0.68372
Proj_2nd_uncal	2407	1023	1005	0.70175	0.70545	0.70360
Proj_2nd_cal	**2421**	**1009**	**991**	**0.70583**	**0.70955**	**0.70769**
Img_3rd	2459	971	953	0.71691	0.72069	0.71880
Proj_3rd_uncal	2581	849	831	0.75248	0.75645	0.75446
Proj_3rd_cal	**2624**	**806**	**788**	**0.76501**	**0.76905**	**0.76703**
Img_cub	2429	1001	983	0.70816	0.71190	0.71003
Proj_cub_uncal	2598	832	814	0.75743	0.76143	0.75943
Proj_cub_cal	**2664**	**766**	**748**	**0.77668**	**0.78077**	**0.77872**

The final output of lane points in the image is up-sampled to the original resolution of CULane to compute AME. The results are shown in Table 3, where the prefix "Img_" and "Proj_" represent fitting in the image and the projected ground, respectively, and the suffix "2nd", "3rd" and "cub" separately present the fitting model of 2^{nd}, 3^{rd} order polynomial, and cubic spline. The results show that in all cases, fitting in the projected space leads to superior results, especially when simple 2^{nd} and 3^{rd} order polynomials are used.

We further compare the resulting F1 measure for different fitting models. As shown in Table 4, no matter what curving models are used, fitting in the projected ground space achieves better performance than in the image space. This is due to the better capability of outlier rejection when fitting the curves in the projected ground plane.

Some qualitative results are shown in Fig. 7. The predicted lanes from SCNN (marked with green points in the left column) are first projected into the ground using the predicted homographic matrix $\boldsymbol{H}$ (blue solid lines in the right column). Then, a group of polynomial curves or cubic splines (red dot lines) are fitted upon these projected lane points. After that, we project the fitted curves back to the original image as the final results (marked yellow). Thanks to the fitting in the projected ground plane, most of the outliers are removed and the position accuracy of lane marks is improved.

4.3.3 Verification on our own dataset

In this section, we evaluate the model on our own dataset, where the camera is calibrated in advance and the full three degrees of rotation is predicted by HP-Net. Some rectified images and their respective projection results are shown in the same column of Fig. 8. Yellow lane points sampled with an equal interval in y-direction become red lane points after using the predicted projection. In the bottom BEV-like images, lane marks become parallel with each other and lane points become uneven(nearby points are dense, distant points are sparse), which conforms to the situations in the real world. Thus, the correctness of the proposed model output is intuitively verified.

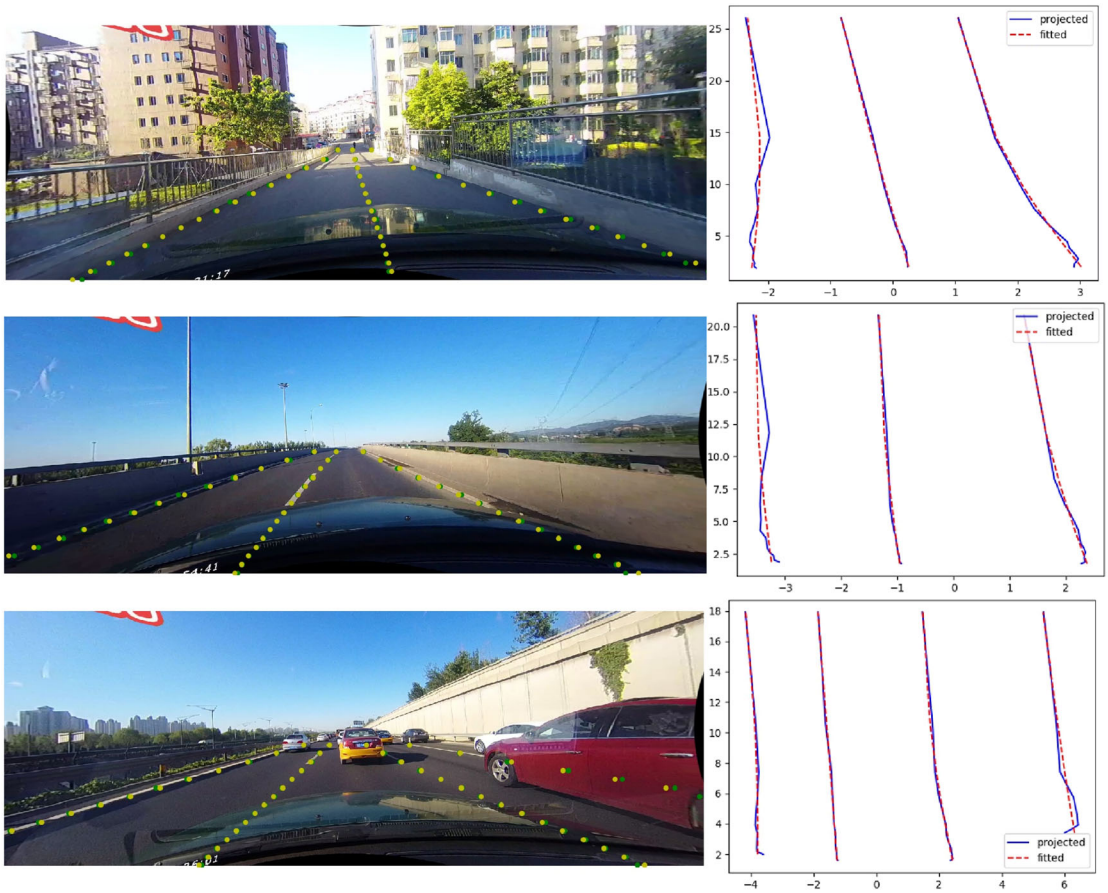


Fig.7 Qualitative results on CULane dataset. Left column: original images with detected lane points (*green*) and re-projected points (*yellow*). Right column: projected lanes (blue solid lines) and 2-order polynomial fitting curves (red dot lines) in the ground

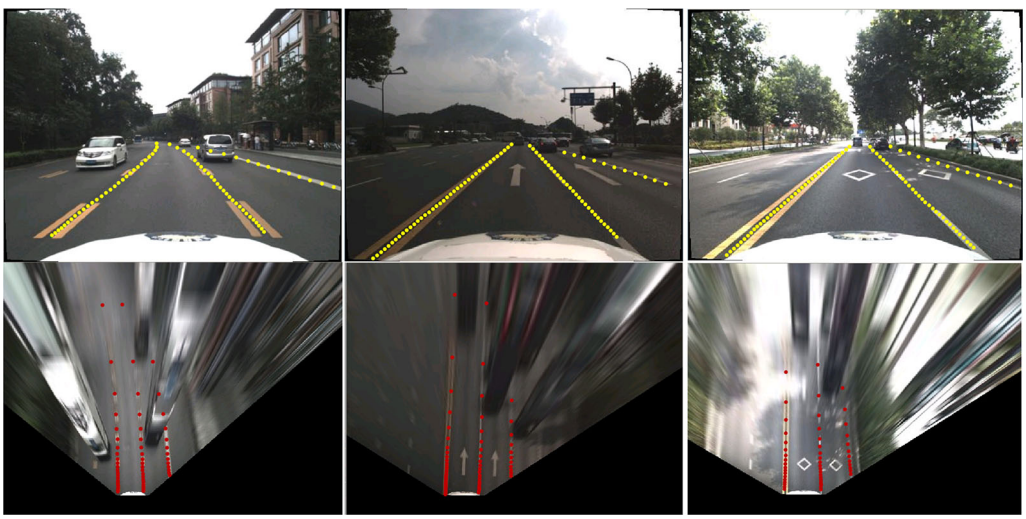


Fig.8 Examples of the projection results on our own dataset. *Top rows*: the acquired images after rectification. Yellow points represent the original sampled lane points. *Bottom rows*: ground plane projection by the predicted homographic matrix. Red points denote the projected lane points

Similar to the evaluations in CULane, F1 metrics are used for our own dataset and the results are shown in Table 5. We can observe that in all cases fitting lanes in the predicted ground plane achieves a higher F1 value than in the original image space. To further verify the actual gain of the camera calibration, we compare them with the uncalibrated model as well. The results show that the projection using calibrated camera (denoted by suffix "cal") model works better than the

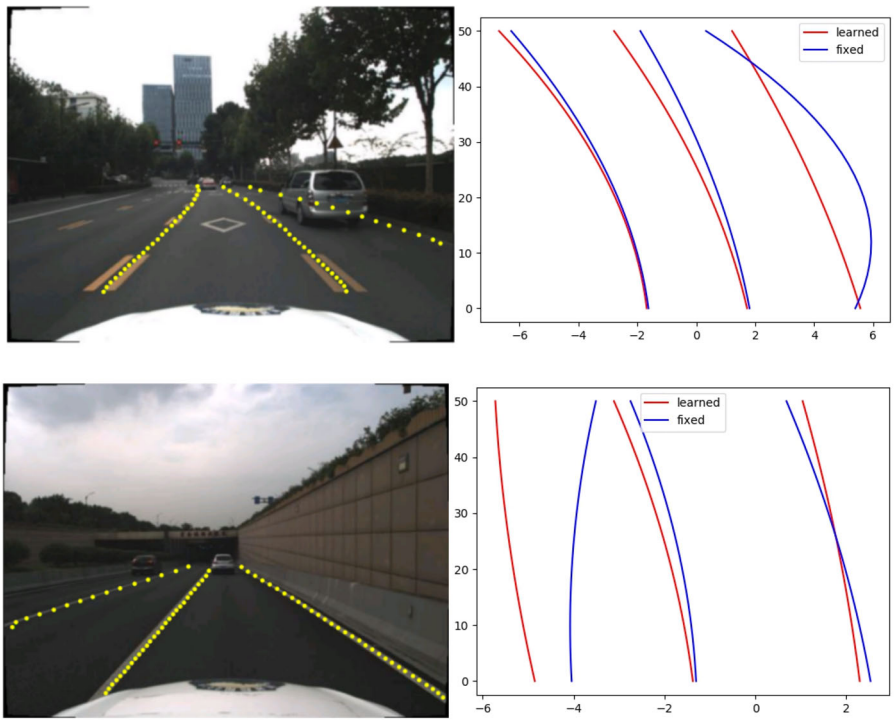


Fig.9 Curve fitting results by the fixed (calibrated once) and the predicted projection in sloping roads. The yellow points in left images are the detection results in the image. The red and blue lines shown in the right column are the fitted lane marks in the ground by the fixed and the predicted projection, respectively

uncalibrated (denoted by suffix "uncal"). This is reasonable because more parameters are known in the calibrated case and the network outputs all 3 degrees of freedom of the rotation.

Some qualitative results are further demonstrated in Fig. 9. We can observe that the fitted lane curves in the predicted ground plane are more parallel and equally spaced than using homographic projection with the fixed (pre-calibrated) parameters. This is more consistent with the real cases, where the lanes are usually parallel to each other and the distance between them is equal. When navigating an autonomous vehicle, accurate lane positions in the ground frame are very important because it can be directly used in lane tracking mode for the vehicle.

To further quantitatively evaluate the projected lane curve's parallelism and equality of distance, we carry out another quantitative evaluation. Given the points projected by the fixed (pre-calibrated) or learned homographic matrix, 2nd order polynomials are used for fitting, and then the curves are sampled every 2 m along the y-direction within the range of 20, 30 and 40 m, respectively. As illustrated in Fig. 10, the lane width $\triangle$X is sampled along the y direction for every 2 m.

For statistics and fair comparison, images without obvious merging and splitting branches are selected out for this evaluation. First, the mean and standard covariance of $\triangle$X for each lane are calculated with:

$$\mathrm{W}_l = \frac{1}{n}\sum_{i=1}^{n}\Delta x_i \tag{12}$$

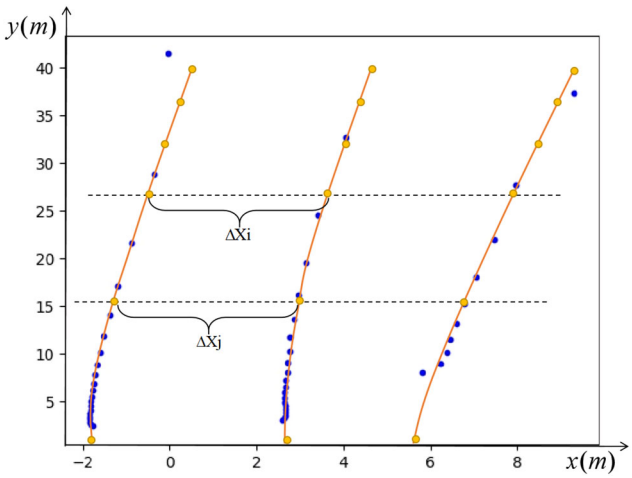


Fig.10 Illustration of measurement of parallelism and distance between fitted lane curves. The projected lane points form the image are marked blue and the fitted lane curves are in orange

$$\mathrm{W\_std}_l = (\frac{1}{n}\sum_{i=1}^{n}(\Delta x_i - W_l)^2)^{1/2} \tag{13}$$

where n is the number of samples on the lane, W_l and $\mathrm{W\_std}_\mathrm{l}$ are the mean and standard deviation of the width for the lth lane, respectively. Given an image with K lanes, the mean width $\overline{\mathrm{W}}$ and the mean difference of width E over the image can be computed as:

Table 6 Comparison of lane's parallelism and width equality error with the fixed (pre-calibrated) or learned projection

Sampling Range (m)	Projection	PE(m)	MAE(m)
[0,2,20]	fixed	0.23571	0.50888
	learned	**0.22070**	**0.40506**
[0,2,30]	fixed	0.42226	0.57595
	learned	**0.40336**	**0.48781**
[0,2,40]	fixed	0.72114	0.73834
	learned	**0.69894**	**0.67366**

$$\overline{\mathrm{W}} = \frac{1}{K}\sum_{l=1}^{K} w_l \tag{14}$$

$$\mathrm{E} = \frac{1}{K}\sum_{l=1}^{K} |w_l - \overline{w}| \tag{15}$$

Finally, the mean absolute error of the width (MAE) and the parallelism error (PE) of the lanes over the entire testing dataset are defined as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_i \tag{16}$$

$$\mathrm{PE} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{l=1}^{K} W\_std_{ik} \tag{17}$$

where K and N are the number of lanes in each image and the number of images in the dataset, respectively. In short words, MAE and PE separately represent the extent of width equality and lane parallelism in the projected ground plane. The smaller these errors are, the better the homographic prediction is. The results are listed in Table 6, where we can see that under all cases using the predicted projection produces better results than using a fixed projection.

Figure 11 illustrates the histogram of MAE in the range of 0 ~ 40 m. We can see that for the predicted projection the error peak lies in the area near zero, while for the fixed projection the peak appears in the range of 0.4 to 0.6 m. It further proves that using the learned projection can achieve more parallel and equal width lane, which is helpful for vehicle navigation.

The computation speeds of the proposed method on different datasets are shown in Table 7. The proposed lane mark fitting module is implemented on a GTX 1080Ti GPU and a mobile computing platform NVIDIA Xavier NX, respectively. It can achieve high efficiency on both testing datasets.

5 Discussion

Despite satisfactory results obtained by our HP-Net, we do observe some failure cases when fitting the lane points in the projected ground plane. Most of the noticeable failures are caused by the incorrectly predicted probability map, which is produced by SCNN in the experiment. In other cases, parallel lanes may be projected incorrectly due to some practical factors. In Fig. 12a, car occlusion causes the leftmost line to be not parallel with the others. Figure 12b shows a phenomenon that some abnormal lane points would be projected too far to be fitted within a reasonable distance. As illustrated in Fig. 12c, inaccurate lane mark candidates can also be caused by the actual uneven road surface. Although these projection situations are less appropriate, the fitting process can still be successfully carried out and normal lane points can be obtained after projecting back into the image space, as shown in the right column of Fig. 12.

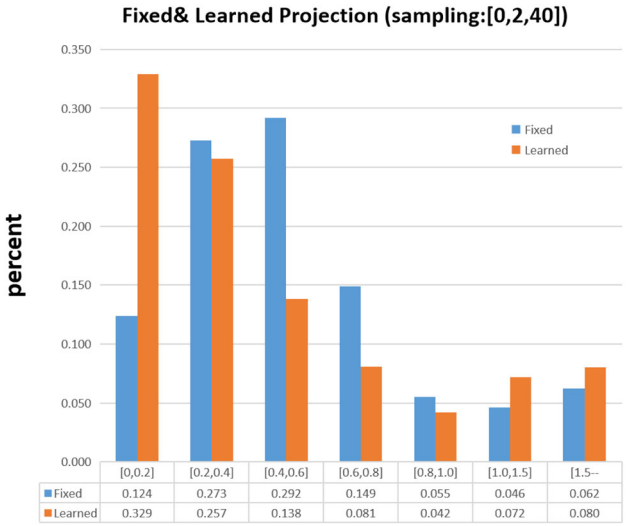


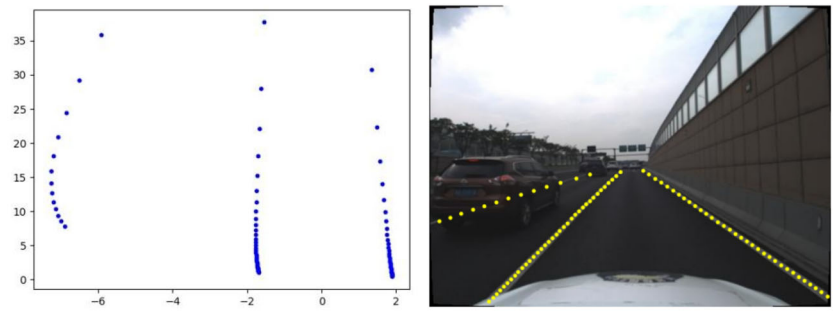
Fig.11 The histogram of MAE for the fixed and learned projection

Table 7 Running speeds (frame rates) of our lane mark fitting module

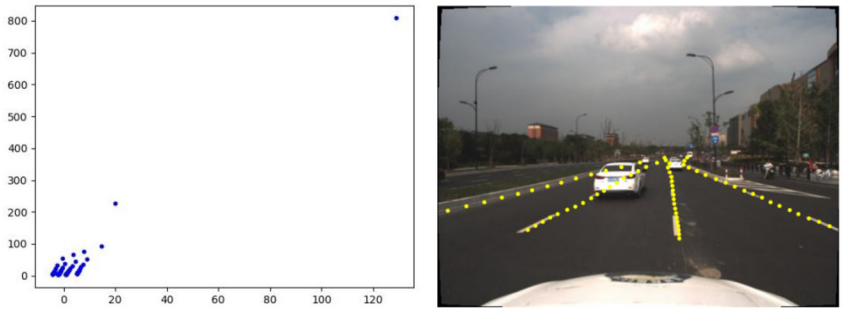
Dataset	GTX (FPS)	NX (FPS)
CULane(128 × 64)	285.7	82.5
Ours(512 × 384)	50	15

6 Conclusion

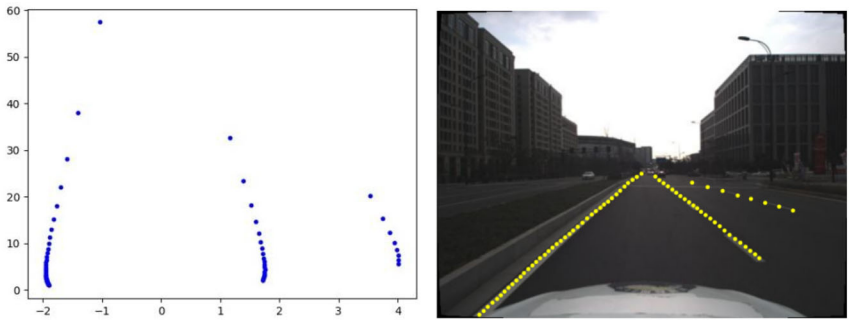
In this work, the HP-Net for adaptively predicting the homographic projection between the image and the sloping ground plane is proposed. By projecting the detected lane mark candidates onto the predicted ground, the lane curves can be optimally fitted with better outlier removal. The detailed pro-



(a) Unparallel lanes caused by occlusion



(b) Existence of distant lane points



(c) Unparallel lanes caused by cuving road surface

Fig.12 Some typical examples of inappropriate projection. The blue dots in the left column are the fitted lane marks in the projected ground plane. The corresponding final detection results are marked by yellow dots in the right image aside

jection model for both calibrated and uncalibrated cameras is presented. Exploring the parallelism nature of the lanes, the network features the capability of being trained by reusing the lane annotations originally for the segmentation task. The existence of a small part of non-parallel lane samples in the training datasets does not affect the convergence of training. Combined with lane segmentation network SCNN, a complete lane detection pipeline is designed. During testing, the lane candidates are fitted separately in the projected ground plane in consideration of possible merging or splitting lanes. The quantitative and qualitative experimental results demonstrate that superior detection performance is achieved by introducing this homography prediction CNN.

Acknowledgments The work is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under grant No.U1709214 and Key Research & Development Plan of Zhejiang Province (2021C01196).

Authors contributions Yiman Chen and Zhiyu Xiang contributed to the study conception and design. Data collection and material preparation were conducted by Yiman Chen and Wentao Du. Experiment implementation and result analysis were performed by Yiman Chen. The manuscript was written by Yiman Chen and all authors approved the version to be submitted.

Funding The work is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under grant no.

U1709214 and Key Research & Development Plan of Zhejiang Province (2021C01196).

Availability of data and material All data and materials support their published claims and comply with field standards.

Code availability Custom code.

Declarations

Conflicts of interest Not applicable.

Ethics approval This work is original and our paper has not been submitted to any other journals. The results are presented clearly, honestly and without fabrication, falsification, or inappropriate data manipulation (including image-based manipulation). We adhere to discipline-specific rules for acquiring, selecting, and processing data.

Consent to participate Not applicable.

Consent for publication All authors agreed with the content and gave explicit consent to publish the paper.

References

1. Chiu, K.Y., Lin, S.F.: Lane detection using color-based segmentation. In: IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, pp. 706–711 (2005)
2. Lee, C., Moon, J.H.: Robust lane detection and tracking for real-time applications. IEEE Trans. Intell. Transp. Syst. **19**(12), 4043–4048 (2018)
3. Liu, G., WÃűrgÃűtter, F., MarkeliÄĞ, I.: Combining statistical hough transform and particle filter for robust lane detection and tracking. In: 2010 IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, pp. 993−997 (2010)
4. Borkar, A., Hayes, M., Smith, M.T.: Robust lane detection and tracking with ransac and kalman filter. In: IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, pp. 3261–3264 (2009)
5. Zhang, F., He, F.: DRCDN: learning deep residual convolutional dehazing networks. Vis. Comput. **36**(9), 1797–1808 (2020)
6. Yang, S., Chen, H., Xu, F. et al: High-performance UAVs visual tracking based on siamese network. Vis. Comput. (2021).
7. Das, D.K., Shit, S., Ray, D.N. et al: CGAN: closure-guided attention network for salient object detection. Vis. Comput. (2021).
8. Kim, J., Lee, M.: Robust lane detection based on convolutional neural network and random sample consensus. In: International Conference on Neural Information Processing (ICONIP), Kuching, Malaysia, pp. 454–461 (2014)
9. Li, J., Mei, X., Prokhorov, D., et al.: Deep neural network for structural prediction and lane detection in traffic scene. IEEE Trans Neural Netw Learn Syst (TNNLS) **28**(3), 690–703 (2017)
10. Lee, S., Kim, J., Shin, Y.J., et al.: Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 1947–1955(2017)
11. Pan, X., Shi, J., Luo, P., et al.: Spatial as deep: Spatial cnn for traffic scene understanding. In: Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 7276–7283 (2018)
12. Loose, H., Franke, U., Stiller, C.: Kalman particle filter for lane recognition on rural roads. IEEE Intell. Vehicles Symp. (IV), pp. 60–65 (2009)
13. Aly, M.: Real time detection of lane markers in urban streets. IEEE Intell. Vehicles Symp. (IV), pp. 7–12 (2008)
14. Gackstatter, C., Heinemann, P., Thomas, S., et al.: Stable road lane model based on clothoids. Adv. Microsyst. Auto. Appl., pp. 133–143 (2010)
15. Ammu, M.K., Philomina, S.: Review of lane detection and tracking algorithms in advanced driver assistance system. Int J Comput Sci Inf Technol **7**(4), 65–78 (2015)
16. Ding, D., Lee, C., Lee, K.Y.: An adaptive road roi determination algorithm for lane detection. In: IEEE International Conference of IEEE Region 10 (TENCON 2013), pp. 1–4 (2013)
17. Chanawangsa, P., Chen, C.W.: A new color-based lane detection via gaussian radial basis function networks. In: International Conference on Connected Vehicles and Expo (ICCVE), Beijing, China, pp. 166–171(2012)
18. Srivastava, S., Lumb, M., Singal, R.: Improved lane detection using hybrid median filter and modified hough transform. J Adv Res Comput Sci Softw Eng **4**(1), 30–37 (2014)
19. Neven, D., De, B.B., Georgoulis, S., et al.: Towards end-to-end lane detection: an instance segmentation approach. IEEE Intell Vehicles Symp (IV), Changshu, China, pp. 286–291 (2018)
20. Hsu, Y.C., Xu, Z., Kira, Z., et al.: Learning to cluster for proposal-free instance segmentation. In: International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, pp. 1–8 (2018)
21. Qu, G., Zhang, W., Wang, Z., et al.: Stripnet: Towards topology consistent strip structure segmentation. In: ACM Multimedia Conference, 283–291 (2018)
22. Li, X., Li, J., Hu, X., et al.: Line-cnn: End-to-end traffic line detection with line proposal unit. IEEE Trans. Intell. Transp. Syst. **21**(1), 248–258 (2020)
23. Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection. Eur. Conf. Comput. Vis. (ECCV), pp. 276–291 (2020)
24. Jongin, Son, Hunjae, et al.: Real-time illumination invariant lane detection for lane departure warning system. Exp. Syst. Appl., Elsevier, **42**(4), 1816–1824(2015)
25. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using b-snake. Image and Vision Computing, Elsevier **22**(4), 269–280 (2004)
26. Tabelini, L., Berriel, R., PaixÃčo, M.: Polylanenet: Lane estimation via deep polynomial regression. In: International Conference of Pattern recognition (ICPR), pp. 1–7 (2020)
27. Liu, R., Yuan, Z., Liu, T., et al.: End-to-end lane shape prediction with transformers. Int Workshop Appl Comput Vis (WACV), pp. 3694–3702 (2021)
28. He, B., Ai, R., Yan, Y., et al.: Accurate and robust lane detection based on dual-view convolutional neutral network. IEEE Intell Vehicles Symp (IV), Gothenburg, Sweden, pp. 1041–1046 (2016)
29. Bruls, T., Porav, H., Kunze, L., et al.: The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping. In: IEEE Intelligent Vehicles Symposium (IV), Paris, France, pp. 302–309 (2019)
30. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: International Conference on Neural Information Processing Systems (NIPS), pp. 2017–2025 (2015)
31. Abadi, M., Agarwal, A., Barham, P.: Tensorflow: Large scale machine learning on heterogeneous distributed systems. arXiv preprint: arXiv:1603.04467(2016).
32. Hou, Y., Ma, Z., Liu, C., et al.: Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, pp. 1013–1021 (2019)
33. Yoo, S., Lee, H., Myeong, H., et al.: End-to-end lane marker detection via row-wise classification. In: CVPR Workshops (2020)

34. Ko, Y., Jun, J., Ko, D., et al.: Key points estimation and point instance segmentation approach for lane detection. arXiv preprint: arXiv:2002.06604(2020)
35. Xu, H., Wang, S., Cai, X., et al.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: European Conference on Computer Vision (ECCV), Glasgow, UK, pp. 1–16 (2020).
36. Tabelini, L., Berriel, R., Paixo, T.M., et al: Keep your eyes on the lane: real-time attention-guided lane detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, pp. 1–9 (2021).
37. Su, J., Chen, C., Zhang, K., et al: Structure Guided Lane Detection. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Canada, pp. 1–8 (2021)
38. Qu, Z., Jin, H., Zhou, Y., et al: Focus on Local: Detecting Lane Marker from Bottom Up via Key Point. arXiv preprint: arXiv:2105.13680(2021)
39. Liu, L., Chen, X., Zhu, S., et al: CondLaneNet: a Top-to-down lane detection framework based on conditional convolution. arXiv preprint: arXiv:2105.05003(2021)

Yiman Chen received her B.E. degree in Electronic Information Engineering from Nanjing University of science and technology, Nanjing, China, in 2017. She is currently pursuing the Ph.D. degree in the College of Information Science and Electronic Engineering at Zhejiang University, Hangzhou, China. Her research interests include computer vision, vision-based navigation, and deep learning.

Zhiyu Xiang received his B.S. and M.S degrees in mechanical engineering and automation and Ph.D. degree in information and telecommunication engineering from Zhejiang University, Hangzhou, China, in 1996, 1999, and 2002, respectively. He is currently a Professor with the College of Information and Electronic Engineering, Zhejiang University. He was a Postdoctoral Research Fellow with the Intelligent Transportation Research Center, The Ohio State University, Columbus, OH, USA, from 2003 to 2004. He is a member of IEEE. His research interests include 2D and 3D computer vision, vision-based navigation, and intelligent transportation systems.

Wentao Du received his B.E. degree in Information Science from Zhejiang University, Hangzhou, China, in 2019. He is currently pursuing a master's degree in the College of Information Science and Electronic Engineering at Zhejiang University, Hangzhou, China. His research interests include computer vision and autonomous driving technology.