A Fast Lane Detection Method Based on Single View Virtual Plane

Zhiming Hou, Yuanzhouhan Cao, Jing Ye

Abstract

Robust lane detection in real-time is one of the foundations for advanced autonomous driving, which can provide substantial amounts of useful information for Autonomous Driving Systems (ADS), vehicle self-control, localization, and map construction. Since the in/extrinsic parameters of different vehicles are various which has a significant impact on the results of 3D lanes, we introduce the Virtual Camera that unifies the in/extrinsic parameters of cameras mounted on different vehicles to guarantee the consistency of the spatial relationship among cameras. It can effectively promote the learning procedure due to the unified visual space. Different from the method of using fixed intrinsic parameters and extrinsic parameters to transformer front-view images to virtual images, we train a network, termed HNet, that estimates the parameters of an ideal perspective transformation, conditioned on the input image. In order to train FVHNet for outputting the transformation matrix that is optimal for perspective transformation, we construct a loss function referred FVHLoss. In summary, we propose a model which can unify the in/extrinsic parameters of different vehicles. We verify our method on the OpenLane dataset and achieve competitive results.

1. Introduction

As one of the foundation for safety in autonomous driving, many research efforts in lane detection focus on making detection model accurate and robust. Over the past few years, 2D lane detection methods have shown impressive performances [**?**, **?**, **?**, **?**, **?**]. However, due to the lack of depth information, transforming 2D images to 3D space still remains highly challenging.

Fortunately, large-scale datasets with 3D lane annotations [**?**, **?**, **?**, **?**] have been proposed, this has greatly facilitated research efforts in 3d lane line representation and detection [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. These methods use single camera images as input for lane line detection in 3D space to improve the accuracy and robustness of the algorithms in real-world scenarios.

Among these methods for 3D lane line detection, BEV-based methods have received attention from researchers due to the accuracy, robustness, and speed. As the general 2D lane Detection methods, the BEV-based methods [**?**, **?**, **?**, **?**, **?**] also requires the features to be transformed into a uniform coordinate space by Inverse Perspective Transformation (IPM) with the camera in/extrinsic parameters. Typically, homographies are estimated between images by finding feature correspondences in those images. The most commonly used algorithms make use of point feature correspondences, though other features can be used as well, such as lines or conics. However, in the existing dataset, we lack this correspondence. As illustrated in Fig.1(a), A common approach is to average all camera in/extrinsic parameters and generate a rough homography matrix. But since the homography matrix is not learnable, these approaches usually can't achieve the best performance.

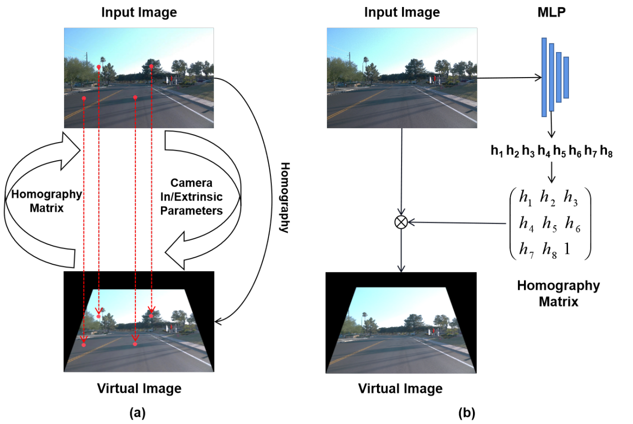

Figure 1. (a) Traditional IPM transformation, which based on camera in/external parameters. (b) DNN-based method, which establish a conditional homography with the specific features of each images.

In order to overcome these problems, we need unify the in/extrinsic parameters of front-facing cameras in different vehicles. Therefore, we propose to extract homography matrix from input images by a MLP network as illustrated in Fig.1(b), which makes homography matrix learnable instead of fixed. As shown in Fig.1(b), HP-Net takes the RGB image as input and predicts the parameters to generate the homography matrix. Then, the input images are projected to the virtual camera by the homography matrix and extracted front-view features by a cnn-based network. We can get the front-view prediction by a front-view lane detection head. Finally, we re-project the front-view prediction 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 without providing any extra annotations except for the original lane labels.

In general, our main contributions are three-fold: 1) HM-Net, a MLP-based network that can learn to predict parameters of the homographic transformation matrix between the input image and the virtual camera image. **2)** Homography Loss, a training approach that only needs the annotations of lane marks, which makes the network to converge more efficiently. **3)** Our method is tested on both the OpenLane dataset and the Apollo 3D synthetic dataset, achieving the promising performance.

2. Related Work

2.1. 2D Lane Detection

2D lane detection [?, ?, ?, ?, ?, ?] aims at obtaining the accurate shape and locations of lanes in the images, and is not concceerned with the spatial extension of the lane line. Earlier works [?, ?, ?, ?, ?, ?, ?] mainly focus on extracting low-level handcrafted features, such as edge and color information. However, these approaches often have complex feature extraction and post-processing designs and are less robust under changing scenarios. In recent years, with the development of deep learning, CNN-based methods have been achieved significant advancements in the field of 2D lane detection. These works are divided into four categories according to pixel-wise segmentation, keypoint-based methods, curve parameters, anchor-based methods.

Segmentation based methods [?, ?, ?, ?, ?, ?] formulate 2D lane detection task as a per-pixel classification problem and typically focus on how to explore more effective and semantically informative features. To make predictions more sparse and flexible, which the computing cost is expensive.

Keypoint-based methods [?, ?, ?, ?] focus on identifying and localizing specific keypoints or landmarks along lane boundaries in images or videos. These methods aim to simplify the lane detection process by directly detecting key features that represent the lanes. While keypoint-based methods can be efficient, they may face challenges in scenarios where keypoints are not well-defined or when there is significant noise or clutter in the image. Additionally, accurately identifying keypoints in varying environmental conditions can be a demanding task.

Curve parameters [?, ?, ?] focus on identifying and characterizing lane boundaries using mathematical curve representations. These methods aim to model the lanes as curves, such as quadratic or cubic functions, and estimate the parameters of these curves to accurately detect and localize the lanes. So it is proposed that the 2D lane detection can be converted into the problem of curve parameter regression by detecting the starting point, ending point, and curve parameters.

Apart from the above methods, **anchor-based methods** [?, ?, ?, ?, ?, ?] design line-like anchors and estimate the offsets between sampled points and predefined anchor points, making them particularly suitable for scenarios with distinct lane patterns. Non-Maximum Suppression (NMS) is then employed to select the lane lines with the highest confidence. LineCNN [?] first defines straight rays emitted from the image boundary to fit the shape of 2D lane lines and applies Non-Maximum Suppression (NMS) to keep only lanes with higher confidence. LaneATT [?] proposes an anchor-based pooling method and an attention mechanism to aggregate more global information. CLRNet [?] learns to refine the initial anchors iteratively through the feature pyramid.

2.2. 3D Lane Detection

Since projecting 2D lanes back into 3D space suffers from inaccuracy as well as less robustness, many researchers have turned their attention to lane detection in 3D space. Unlike traditional 2D methods that operate solely in the image plane, 3D lane detection leverages depth information to provide a more comprehensive understanding of the road environment, enabling vehicles to perceive and navigate lanes in real-world scenarios with varying terrains, elevation changes, and complex road geometries.

Some works restore 3D information using multiple sensor [?, ?, ?]. While 3D lane detection offers significant advantages, it also comes with challenges such as computational complexity, sensor calibration and the collection and annotation cost of multisensor data is expensive. Therefore, monocular camera image based 3D lane detection [?, ?, ?, ?, ?, ?, ?] attracts more attention.

Due to the good geometric properties of lanes in the perspective of BEV, **3DLaneNet** [?] predict the position of lanes in 3D space. It utilizes an Inverse Perspective Mapping (IPM) technique to transform features from a front-view image into a Bird's Eye View (BEV) representation, where the geometric properties of lanes are more easily discernible. By regressing the anchor offsets in the BEV space, 3DLaneNet can accurately predict the position of lanes without relying on the assumption of a flat ground. **Gen-LaneNet** [?] improves the alignment between the virtual top view generated by an inverse perspective mapping (IPM) and the true top view in 3D space. By distinguishing between these views, Gen-LaneNet enhances the accuracy of lane detection without the need for a bird's-eye view (BEV) transformation. **Persformer** [?] utilizes deformable attention to generate bird's-eye-view (BEV) features more adaptively and robustly, improving the accuracy and reliability of 3D lane detection without relying on the flat ground assumption. **SALAD** [**?**] tries to get rid of BEV by decomposing 3D lane detection into 2D lane segmentation and dense depth estimation tasks. **Anchor3DLane** [**?**] predict 3D lanes directly from frontal-viewed (FV), which defines 3D lane anchors in the 3D space and projects them onto the FV features to extract structural and contextual information for accurate predictions, and incorporates a global optimization technique to reduce lateral prediction errors by leveraging the equal-width property between lanes. **BEV-LaneDet** [**?**] establishes a Virtual Camera with standard in/extrinsic parameters to ensure the consistency in the spatial relationship among cameras, and introduces a Spatial Transformation Pyramid module for transforming front-view features into Bird's Eye View (BEV) features.

3. Methods

An overview of our entire lane detection framework is illustrated in Fig.2. Our approach takes a single image captured by a front camera mounted on the vehicle as input information. The image from the current camera is projected onto the view of the virtual camera using the homography matrix generated by MLP network [**?**], which aims to transform the in/external parameters of the input image to a unified in/external parameter of the virtual camera. We use ResNet18 and ResNet34 [**?**] as our backbone to extract front view image features. Spatial Transformation Pyramid [**?**] transform the front view features into BEV features. Then, lanes are predicted on the BEV view. We predict the confidence of each cell, the embedding used for clustering, the offset from the center of the cell to the lane in the y-direction, and the height. we added the front view lane detection header as an auxiliary supervision to improve backbone's ability to extract front view features and supervise the homography matrix estimation network's training.

3.1. Homography

new why use homography 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.

101 011 001 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 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 H. Depending on whether the camera is pre-calibrated, different outputs of the model are designed.

The zeros are placed to enforce the constraint that horizontal lines remain horizontal under the transformation.

old A homography is a projective transformation between two planes or, alternatively, a mapping between two planar projections of an image. In other words, homographies are simple image transformations that describe the relative motion between two images, when the camera (or the observed object) moves. It is the simplest kind of transformation that describes the 2D relationship between two images. Homography can be mathematically described by a 3D transformation in a homogeneous coordinates space and can be expressed as:

$$S\begin{bmatrix}x'\\y'\\1\end{bmatrix}=H\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}h_1&h_2&h_3\\h_4&h_5&h_6\\h_7&h_8&h_9\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

where H denotes the homography matrix $H \in R^{3\times3}$ between two two-dimensional planes, which allows us to switch from one view of the same scene to another by multiplying the homography matrix with the points [u,v] in one view to find their corresponding positions [u',v'] in the other view. The homography matrix is usually parameterized by the elements of a 3×3 matrix, but it has only 8 degrees of freedom, and a simple way to do this is to hardcode $h_9 = 1$. We take the original image as input and use DNN to estimate eight parameters of the homography matrix.

3.2. Unified Camera Parameters

Different cameras mounted on the vehicle have different internal/external parameters, which have a significant effect on the 3D lane results. As shown in Fig.3, By creating a virtual camera with standard internal/external parameters, we can unify the internal/external parameters of different cameras. We project the image of the current camera into the view of the virtual camera through the homography matrix H based on the principle of perspective transformation. As a result, the virtual camera unifies the internal/external parameters of different cameras. We use a MLP network to estimate the 8 parameters of the homography matrix from the input frames. Then, we transform from one view of the same scene to another by multiplying the homography matrix with the points $[u, v]$ in one view to find their corresponding locations $[u', v']$ in the other view.

3.3. Homography Loss

as shown in Fig.4. In order to improve the MLP network's ability to extract homography matrix parameters and backbone's ability to extract front view features, a front view lane detection header was added as an auxiliary supervision. Based on this front view lane detection head, we

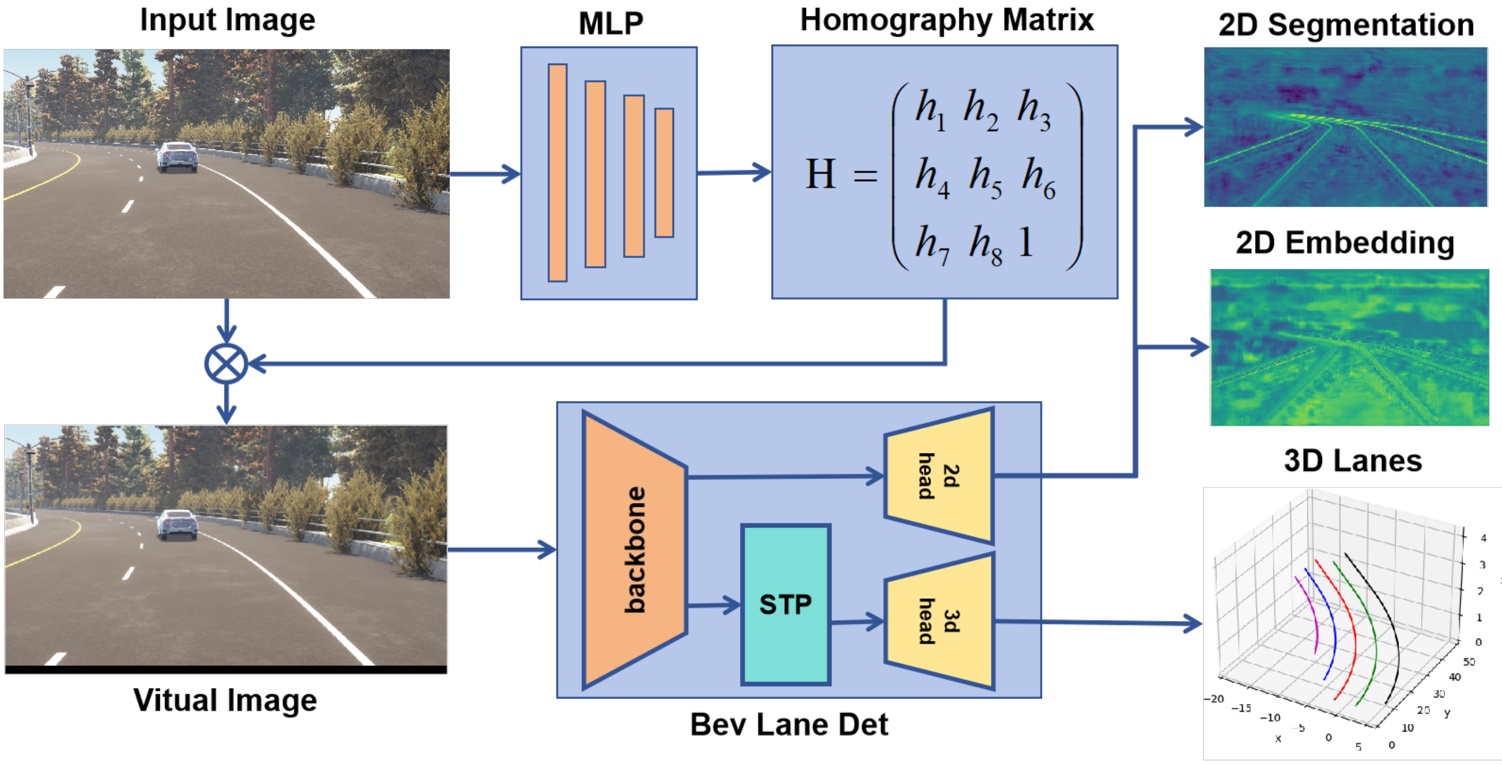

Figure 2. Overview of the framework

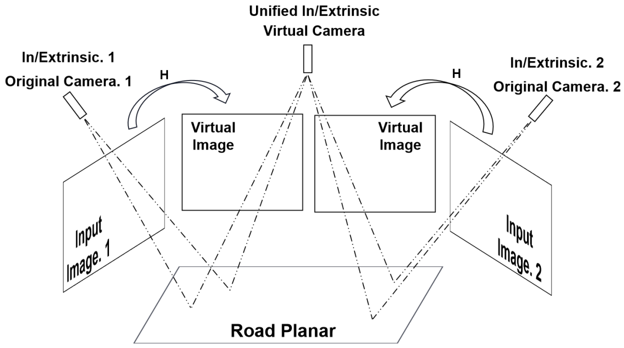

Figure 3. Virtual Camera

propose the homography loss function. After the input image is converted to a virtual camera image by homography matrix H, the front view features of the virtual camera image are obtained by backbone, and the front view lane detection head gets the virtual camera segmentation and embedding results Seg_v, Emb_v, which can be converted back to the input camera Seg_i, Emb_i by the inverse matrix H^{-1} of the homography matrix. The front-view lane loss includes lane segmentation loss and lane embedding loss, referred to LaneNet [**?**].The homography loss is defined as follows,

$$L_h = \lambda^h_{seg}L^h_{seg} + \lambda^h_{emb}L^h_{emb}$$

where L^h_{seg} denotes lane segmentation loss, andL^h_{emb} denotes lane embedding loss in the front-view.

4. Expriments

4.1. Dataset And Experimental Setup

apollo 3D Lane Synthetic Dataset. Apollo Synthetic dataset [**?**] consists of over 10k 1080 × 1920 images which are built using unity 3D engine, including highway, urban, residential and downtown environments. The dataset is split into three different scenes: balanced scenes, rarely observed scenes and scenes with visual variations for evaluating algorithms from different perspectives.

OpenLane Dataset. OpenLane Dataset [**?**] is the first real world 3D lane dataset which consists of over 200K frames at a frequency of 10 FPS based on Waymo Open dataset [32], [33]. In total, it has a training set with 157k images and a validation set of 39k images. The dataset provides camera intrisics and extrinsics following the same data format as Waymo Open Dataset.

4.2. Experiment Settings

Implementation Details. We use ResNet [**?**] as backbone. The resolution of our input image is 576 × 1024. The bev range is set to [3, 103] × [-12, 12] along x and y respectively. Each cell represents x × x (x defaults to 0.5m). Our network

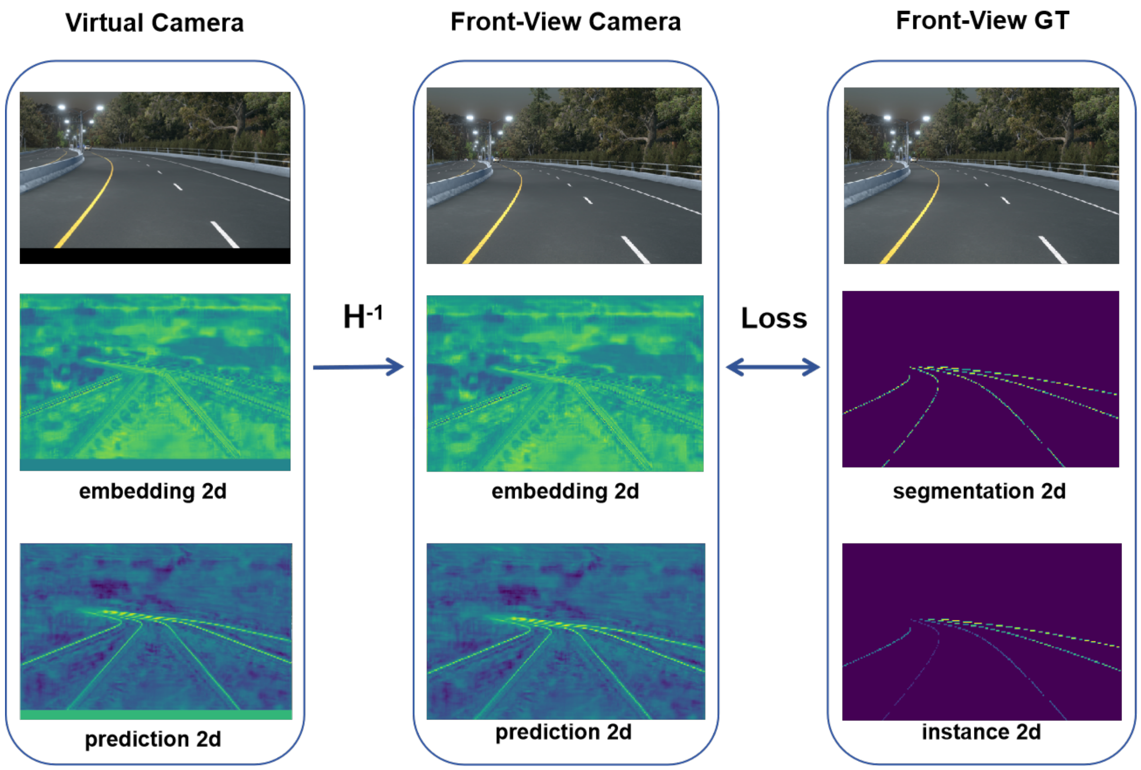

Figure 4. Homography Loss

Table 1. Comparison with previous methods on Apollo 3D Lane Synthetic Dataset.

Scene Method	F-Score	X error near	X error far	Z error near	Z error far	
Balanced Scence	3D-LaneNet [8]	86.4	0.068	0.477	0.015	**0.202**
	Gen-LaneNet [9]	88.1	0.061	0.496	0.012	0.214
	3D-LaneNet(1/att) [13]	91	0.082	0.439	0.011	0.242
	Gen-LaneNet(1/att) [13]	90.3	0.08	0.473	0.011	0.247
	PersFormer [3]	92.9	0.054	0.356	0.01	0.234
	Ours	**97.2**	**0.033**	**0.273**	0.044	0.263
Rarely Observed	3D-LaneNet [8]	72	0.166	0.855	0.039	**0.521**
	Gen-LaneNet [9]	78	0.139	0.903	0.03	0.539
	3D-LaneNet(1/att) [13]	84.1	0.289	0.925	0.025	0.625
	Gen-LaneNet(1/att) [13]	81.7	0.283	0.915	0.028	0.653
	PersFormer [3]	87.5	0.107	0.782	0.024	0.602
	Ours	**95.5**	**0.059**	**0.646**	0.090	0.685
Vivual Variants	3D-LaneNet [8]	72.5	0.115	0.601	0.032	**0.23**
	Gen-LaneNet [9]	85.3	0.074	0.538	0.015	0.232
	3D-laneNet(1/att) [13]	85.4	0.118	0.559	0.018	0.29
	Gen-LaneNet(1/att) [13]	86.8	0.104	0.544	0.016	0.294
	PersFormer [3]	89.6	0.074	0.43	0.015	0.266
	Ours	**94.3**	**0.051**	**0.389**	0.043	0.307

uses Adam optimizer [**?**], with a base learning rate of 10^{-3} and weight decay of 10^{-2}. All models are trained from scratch with 120 epochs and the per-GPU batch size is set to 8.

4.3. Evaluation

The evaluation metrics we used are referred from Gen-LaneNet [?], which includes F-Score in different scenes and X/Z error in different regions. We report Average Precision (AP) , Fscore, and errors (near range and far range) to investigate the performance of our model.

4.4. Comparisons with existing methods

5. Conclusions

conclusion conclusion conclusion