# Research on the Lane Recognition Method Based on Computer Vision

**Weiqi Kong**

Jiangxi Normal University (Yaohu Campus), 99 Ziitao Dadao, Nanchang City, Jiangxi Province, China

E-Mail: 1678637972@qq.com

**Abstract.** The lane line is the most important traffic sign in road traffic and plays a significant function in restricting and guaranteeing the running of vehicles. Whether in the vehicle safety driving system or in the intelligent vehicle navigation based on machine vision, lane detection and recognition is a basic and necessary function module. This can enable future in-depth studies on intelligent transportation while also lowering the likelihood of traffic accidents.

**Keywords:** lane recognition, image semantic segmentation network, geometric detection, anchor representation, image segmentation.

## 1. Background

In 2020, Ren and his group came up with a proposal for an end-to-end lane detecting methodology. This model is divided into two parts: the first part is feature extraction by deep network, that is, a weight is given to each pixel in the picture to represent the likelihood that the pixel is a member of the lane line; in the second part, based on the weight extracted in the first part and the curve in ground truth, the network parameters are modified through back propagation to output the parameters fitting the lane line curve. They implemented lane line detection in an end-to-end manner, and two parts of the network implemented the recognition weight and curve models, respectively. From the experimental effect, their model completely identified the lane lines [1]. In 2020, a team of researchers at the University of Michigan in Ann Arbor proposed a real-time enhanced lane detection method for understanding scene physics. The model is divided into two parts: the first component is a hierarchical semantic segmentation network, which functions as a scene feature extractor; the second part is lane inference using a physically improved multi-lane parameter optimization module. Semantic segmentation in this model relates to the method of giving each pixel in a picture a semantic label. Types of labels include Cityscape and Vistas [2]. In the literature [3], researchers learn a richer structure and context through the network built by the transformer, where the lane shape model is developed based on the road structure and camera posture, which can provide a physical interpretation for the network output parameters. Transformer uses a self-attention mechanism to model non-local interactions to capture their slender structure and global context. In the literature [4], researchers used ultrafast structure-sensing lane lines to detect the new network. The lane detection process is treated as an issue with row-based selection using global characteristics to select lane locations in the predetermined semantic rows of the image rather than each pixel of the lane segmentation based on the

local receiving domain. In addition, they also proposed structural loss to model the channel structure. In the literature [5], researchers proposed a panoramic driving awareness network to simultaneously segment the drivable area, detect lane lines, and detect traffic targets. The team came up with the YOLOP model. The model consists of three decoders for processing particular jobs and an encoder for feature extraction. Encoders are consisting of a neck network and a backbone network. The decoder consists of three heads: the detection head, the feasible domain head, and the lane line head. In the literature [6], researchers proposed a semantic segmentation method for lane markings based on the fusion of lidar and cameras. In order to obtain accurate location information in the segmentation results, the semantic segmentation object of this method is the converted aerial view from the LIDAR point cloud rather than the image captured by the camera. First, researchers use the network to segment the captured images and then combine the segmentation results with the point cloud collected by LIDAR as the input of the network. They also added long- and short-term memory structures to help the network segment lanes semantically by using time series information. In literature [7], the research team of Huawei and Sun Yat-sen University proposed the lane-sensitive architecture shrinking framework. The framework comprised in three sections: the first section is the feature fusion search module, which is used to better integrate local and global contexts for the features of the multi-layer jagged structure; the second part is the elastic trunk search module, which explores the feature extractor with good semantic information and potential information. The third part is an adaptive point mixing module, which is used to search for multi-level post-processing strategies to combine the prediction results of multi-scale heads. In the literature [8], researchers proposed a generic, scalable, and 3D lane detecting technique. This method proposes to introduce a new representation of geometric lane lines in a new system of coordinates and straight from the network output, apply a certain geometric modification to determine genuine 3D lane positions. Second, they propose a scalable two-stage architecture that separates learning from geometric coding subnetworks and image segmentation subnetworks.

The technique that the author of this article employed consisted of first projecting the three-dimensional scene onto the image plane by using projection transformation, then projecting the image that was captured onto the flat road plane by using plane homography, and finally projecting a point of the panoramic view onto the same image pixel, which had to be on the same light. It was decided to make the optical center of the camera the starting point for the camera's coordinate system, and the vertical projection point from the camera to the ground was taken as the system of coordinates for the automotive body and the origin of the panoramic view. Then, the image was encoded by the network, the features were converted into the panoramic view, and the network was also utilized to make predictions about the lane locations that were represented in the panoramic view. In this paper, the 3D lane locations in the coordinate system for the car body are calculated by the author using geometric transformation.

## 2. Content of This Research Project

### *2.1. Geometric Detection of 3D Lane Lines*

The x, y, and z axes and the origin O represent the car body coordinate system, perpendicular to the road; xc, yc, and zc, and the origin C, represent the camera coordinate system. Therefore, it is possible for us to construct the view by first projecting the three-dimensional scene onto the image plane using projection transformation, and then projecting the picture that was acquired onto the flat road plane using plane homography. Since camera parameters are involved, points in the panoramic view correspond to corresponding 3D points in the car body coordinate system and have different x, y, and z values in principle. Let's derive the relationship between the panoramic view coordinates and the real-world coordinates. While designing a projection camera, it is important to ensure that the 3D point (x, y, z), its projection on the image plane, and the optical center of the camera (0, 0, h) are all positioned on a single ray. Accordingly, if a point of the panorama view is projected onto the same image pixel, that point must be on the same ray. Therefore, the center of the camera (0,0, h), the 3D

point (x, y, z), and their corresponding panoramic view points appear collinear, and the relationship between these three points can be written as:

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

Taking the camera's optical center as the camera coordinate system's origin and the vertical projection point from the camera to the ground as the automotive body coordinate system's origin and panoramic view, we derive the conversion relationship between the coordinates of the God view chart and the coordinates of the automotive body coordinate system. According to the proposed geometric shape, 3D lane detection was solved in two steps. First, we encoded the image using the network in order to turn the features into a panoramic view and to make a prediction about the lane locations that were represented in the panoramic view. Then we use a geometric transformation to calculate 3D lane points in the carbody coordinate system. Specifically, the steps are as follows:

- Step 1: Use an image semantic segmentation network to predict lane masks.
- Step 2: Transform the mask into a panoramic view using the Inverse Perspective Mapping (IPM) module and the camera's internal parameter matrix.
- Step 3: Predict lane lines in the panoramic view.
- Step 4: Map the lanes in the panoramic view back to real-world coordinates using the above geometric relationships.

Similar to the 3D lane line network, the anchor representation enables the network to directly predict 3D lane lines in the form of multiple lines. The essence of anchor representation is to use a network to realize contour grouping and boundary detection in a structured scene. The network that was employed for this study produces 3D lane lines in the panoramic view based on this anchor representation, and then it applies the transformation algorithm that was derived earlier to calculate the 3D lane points that correspond to those lane lines. Taking into account each lane point's projected likelihood of visibility, reserve only those lane points that are visible and have a high probability of contributing to the final output. It has two main features: (1) Anchor position: We define N equidistant vertical lines of x position and y position in advance; (2) values to be predicted by the anchor: offset, z coordinate of the current point, visibility v, probability p. Compared to 3D lane line networks, anchor representation involves two major improvements: (1) Representation of lane point locations in different spatial panoramas Representing lane points in a panoramic view ensures that the target lane location is aligned with the image features projected into the top view. (2) Unlike the 3D lane network's global coding of the entire scene, the local patch-level correlation coding adopted by the method used in this paper is more robust when dealing with new or unseen scenes. The approach used in this article adds additional properties to the anchor representation, such as the visibility of each anchor point. Due to this, the method is more stable when dealing with partially visible lane lines that begin or end midway.

The loss function of model training is predicted as follows:

$$L^{tiles}=\sum_{i,j\in W\times H}(L_{ij}^{score}+c_{ij}\cdot L_{ij}^{angle}+c_{ij}\cdot L_{ij}^{offsets}) \tag{2}$$

Among this:

$$L_{ij}^{offsets}=\left\|\tilde{r}_{ij}-r_{ij}\right\|+\left\|\tilde{\Delta}z_{ij}-\tilde{\Delta}z_{ij}\right\| \tag{3}$$

$$L_{ij}^{angle}=\sum_{\alpha=1}^{N_\alpha}\left[p_{ij}^{\alpha}\cdot\log\tilde{p}_{ij}^{\alpha}+(1-p_{ij}^{\alpha})\cdot\log(1-\tilde{p}_{ij}^{\alpha})+\delta_{ij}^{\alpha}\cdot\left\|\tilde{\Delta}_{ij}^{\alpha}-\Delta_{ij}^{\alpha}\right\|\right] \tag{4}$$

$$L_{ij}^{score} = c_{ij} \log \tilde{c}_{ij} + (1 - c_{ij}) \cdot \log(1 - \tilde{c}_{ij}) \quad (5)$$

The discriminative push-pull loss utilized for lane clustering’s global embedding is as follows:

$$L^{embedding} = L^{pull} + L^{push} \quad (6)$$

This includes:

$$L^{pull} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{ij \in W \times H} \left[ \delta_{ij}^{c} \cdot \left\| \mu_c - f_{ij} \right\| - \Delta_{pull} \right]^2 \quad (7)$$

$$L^{push} = \frac{1}{C(C-1)} \sum_{C_A=1}^{C} \sum_{C_B=1}^{C} \left[ \Delta_{push} - \left\| \mu_{C_A} - \mu_{C_B} \right\| \right]^2 \quad (8)$$

In order to create the camera coordinate system, we finally translate the lane line points in the BEV plane:

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \\ \tilde{z}_{ij} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{cam}) & \sin(\varphi_{cam}) \\ 0 & -\sin(\varphi_{cam}) & \cos(\varphi_{cam}) \end{bmatrix} \cdot \begin{bmatrix} r_{ij} \cdot \cos(\tilde{\phi}_{ij}) \\ r_{ij} \cdot \sin(\tilde{\phi}_{ij}) \\ \tilde{\Delta} z_{ij} - h_{cam} \end{bmatrix} \quad (9)$$

*2.2. Geometric Detection of 3D Lane Lines*

As shown in Figure 1, the x, y, and z axes and origin O represent the coordinate system of the vehicle body, perpendicular to the road; and xc, yc, zc, and origin C represent the camera coordinate system. Therefore, it is possible for us to construct a panoramic view by first projecting the three-dimensional scene into the image plane using projection transformation, and then projecting the image that was acquired onto a flat road plane using plane homography. Considering that camera settings are relevant, points in the panoramic view correspond to corresponding 3D points in the car body coordinate system and have different x and y values in principle.

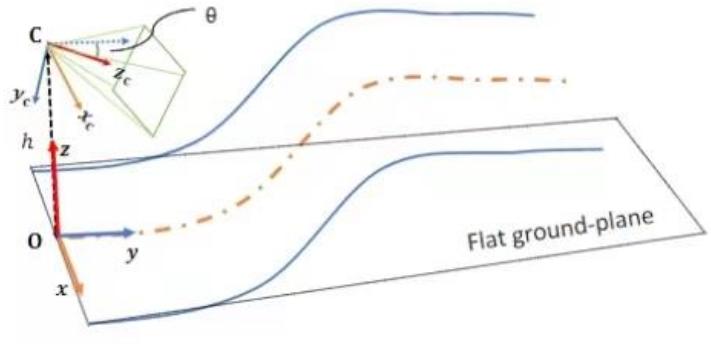


**Figure 1.** Camera setup and ego-vehicle coordinate frame.

We derive the relationship between the panoramic view coordinates and the real-world coordinates. While designing a projection camera, it is important to ensure that the 3D point (x, y, z), its projection on the image plane, and the optical center of the camera (0, 0, h) are all positioned on a single ray. In a similar vein, in order for a point in the panorama view to be projected onto the same image pixel, that point needs to be on the same ray. Therefore, the center of the camera (0,0, h), the 3D point (x, y, z),

and their corresponding panoramic view points appear collinear, which shows in Figure 2 (a) and (b). Formally, the relationship between these three points can be written as follows:

$$\frac{h-z}{z}=\frac{x}{\bar{x}}=\frac{y}{\bar{y}} \quad (10)$$

We take the camera's optical center as the camera coordinate system's origin and the vertical projection point of the camera to the ground as the automotive body coordinate system's origin and the panoramic view. Then the conversion relationship between the coordinates of the panoramic view and the coordinates of the automotive body coordinate system is derived from the following figure:

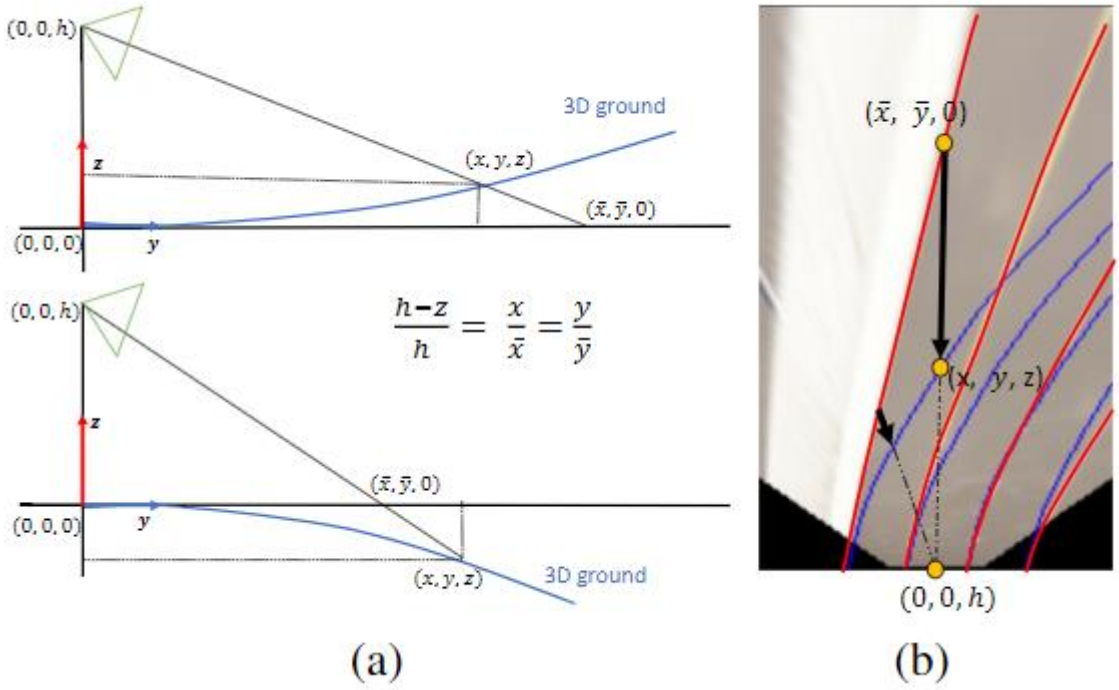


**Figure 2.** Geometry in 3D lane detection.

Whether 0 (top)>or z<0 (bottom), the collinear correlation between 3D lane points (r, 3y, z) and their projection on the virtual top view (0, g, 0) and camera center (0, 0, h) remain unchanged. In the virtual top view, we make the assumption that the lane height z is conceptually similar to the estimate vector field (represented by the black arrow), and then we move the top view lane points (represented by the red curve) to their desired positions so that they can form parallel curves (blue curve).

$$\begin{aligned} x &= \bar{x}\cdot(1-\frac{z}{h}) \\ y &= \bar{y}\cdot(1-\frac{z}{h}) \end{aligned} \quad (11)$$

*2.3. An Anchor Representation of Geometric Guidance*

According to the proposed geometry, our two-step approach to solving the 3D lane detecting: First, we use the network to encode the image, convert the features into a panoramic view, and predict the lane points represented in the panoramic view; secondly, we use geometric transformation to calculate the 3D lane points in the car body coordinate system. Specifically, the steps are as follows:

(1) Use an image semantic segmentation network to predict lane lines.

(2) Use the Inverse Perspective Mapping (IPM) module to convert the lane map into the God view map (requires the camera's internal parameter matrix).

(3) Predict lane lines in the God View diagram.
(4) Map the lane lines in the God's View graph back to real-world coordinates using the geometric relationships derived in the previous section.

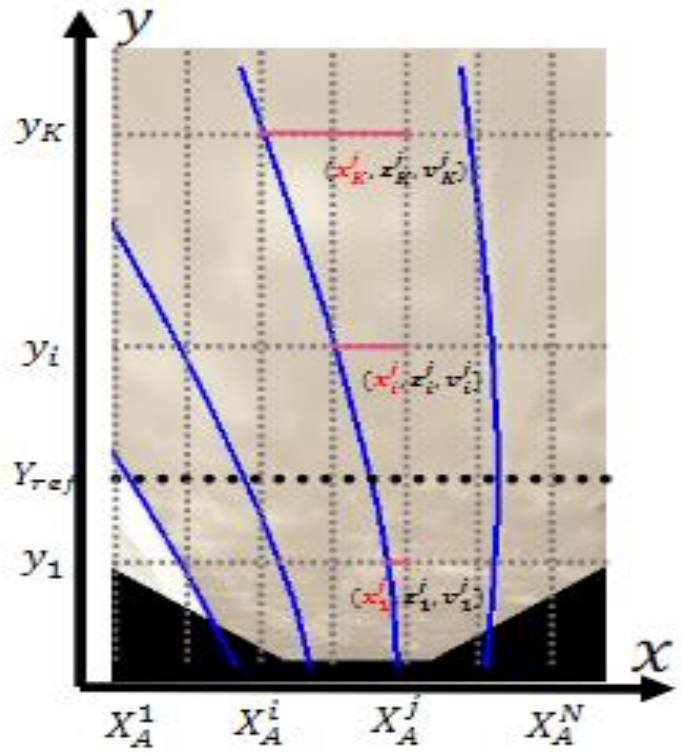


**Figure 3.** Anchor points representation.

Lane anchors are defined as N equidistant vertical lines in x-position {XA}g1. Given a set of predefined fixed y positions {yi}1, 3D lanes can be used by the 3 • K attribute {(yuan, z, v)}.

*2.3.1. On the basis of the x value at Yref, the ground truth lane is connected to its nearest anchor point*

Similar to the 3D lane line network, the anchor representation enables the network to directly predict 3D lane lines in the form of multiple lines. The essence of anchor representation is to use a network to realize boundary detection and contour grouping in a structured scene. According to this anchor representation, the network that was utilized for this study generates 3D lane lines in the panoramic view. Following this step, the transformation method described earlier is used to calculate the 3D lane points that correspond to these lane lines. Given the predicted probability of visibility for each lane point, only those lane points with a high probability of visibility are reserved for the final output. It has two main features: (1) anchor position: N equidistant vertical lines of x position and y position are defined in advance; (2) values to be predicted by the anchor: offset x and z coordinates of the current point, visibility v, and probability p.

Anchor representation involves two major improvements over 3D lane line networks: (1) Representation of lane point locations in a panoramic view: this method ensures that the target lane location is aligned with the image features projected into the top view. (2) Unlike the 3D lane network ’ s global coding of the entire scene, the local patch-level correlation coding adopted by the method used in this paper is more robust when dealing with new or unseen scenes. The approach used in this article adds additional properties to the anchor representation, such as the visibility of each anchor point. Due to this, the method is more stable when dealing with partially visible lane lines that begin or end midway.

*2.4. A Two-Stage Network Framework with Image Coding and Geometric Inference Coupled Learning*

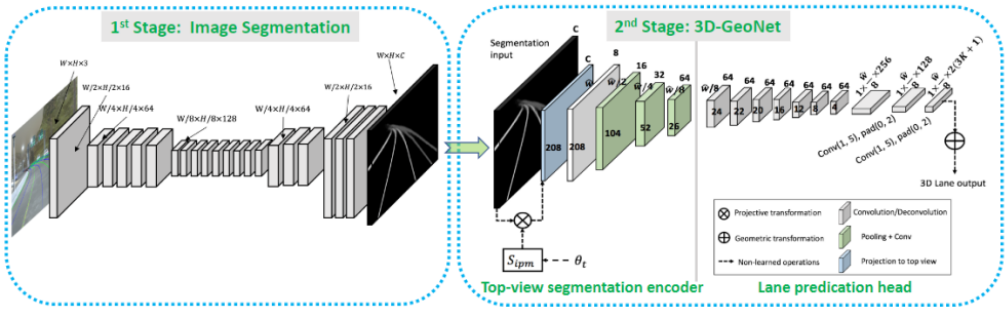


**Figure 4.** Two-Phase network architecture.

In the first stage, we feed input images of size W * H into the image segmentation subnet to generate channel segmentation with the same resolution. The middle segmentation map uses a 3D universal network, which is made up of a top view segmentation encoder and a lane prediction head, to generate 3D lanes that are then represented by a virtual top view. In the final step of this process, we make use of the geometric transformation in order to compute the 3D lane locations within the coordinate system of the car body.

Stage 1: Image semantic segmentation network

Stage 2: 3D universal network Output prediction lane position based on a split network

Using an end-to-end learning network, this paper uses a two-stage framework to decouple image coding learning and 3D geometric extrapolation understanding. As can be seen in Figure 4, the primary objective of the first subnetwork is the classification of lane lines inside the image domain. The segmented output from the first subnetwork is used as the input for the second subnetwork, which then makes a prediction about the 3D lane structure. The framework of these two stages shows that 3D geometry coding and image feature extraction are independent of each other. As shown in Figure 2(a), the ground height z value is closely related to the displacement vector from the position (x1,y1) to the position (x2,y2). In fact, in the panoramic view, we believe that estimating the lane height z is conceptually identical to estimating the black arrows that are part of the vector field and are responsible for moving the red curve of the top view lane point to its final destination position so that blue curves can be formed in parallel. Over the same time period, the network will be encoding the correlation that exists between the visual features and the target vector field. Due to the fact that the target vector field is primarily concerned with geometry, the simple characteristics recovered from sparse lane segmentation are sufficient.

In the first stage, 2D image segmentation can be achieved by using lightweight semantic segmentation network models, such as ERFNET, FCN, etc. For simplicity, we chose to apply ERFNET to the first phase of the framework. For the second phase of 3D lane alignment prediction, we proposed a 3D-3D universal network, as shown in Figure 4. The top-view segmentation encoder works by first projecting the segmentation map input onto the top-view layer, and then encoding the data from the segmentation map into the feature map using a sequence of convolutional layers. The lane line prediction head is responsible for predicting the position of the lane points on the top view as well as the attributes of the 3D lane lines that are based on the anchor representation.

The average end-to-end learning framework is very sensitive to the appearance of images and relies on large amounts of real-world 3D data. However, the two-stage approach in this paper can significantly reduce the cost because it is no longer necessary to gather redundant real 3D lane label data in the same location at different times of day, with varying degrees of occlusion, and different types of weather. In addition, the two-stage framework can be used to train a more reliable two-dimensional lane segmentation subnetwork with more adequate two-dimensional real data. The performance of 3D lane prediction is significantly improved when exceptionally accurate segmentation is used as input. The two-stage framework can, in the best-case scenario, train the image segmentation subnetwork

using 2D real data and train the 3D geometry subnetwork using only synthesized 3D data. This requires domain transfer techniques to resolve the domain gap that exists between the perfect synthetic segmentation base truth value and the segmentation output of the first subnetwork.
The loss function is shown in Formula (12):

$$l = -\sum_{t\in\{c,l\}}\sum_{i=1}^{N}( p_t^i \log p_t^i + (1-p_t^i)\log(1-p_t^i))$$
$$+\sum_{t\in\{c,l\}}\sum_{i=1}^{N}\hat{p}_t^i\cdot(\left\|\hat{v}_t^i\cdot(x_t^i-\hat{x}_t^i)\right\|+\left\|v_t^i\cdot(z_t^i-\hat{z}_t^i)\right\|) \quad (12)$$
$$+\sum_{t\in\{c,l\}}\sum_{i=1}^{N}\hat{p}_t^i\cdot\left\|v_t^i-\hat{v}_t^i\right\|$$

Compared with the loss function introduced in the 3D lane line network, it has three changes. First of all, x belongs to the panoramic frame, not to the body frame. Second, we decided to add an extra loss term so that we could have a better idea of how much of a gap there was between the projected visibility vector and the Real visibility vector. Third, the distance loss term is multiplied by the visibility probability that corresponds to it, v, to ensure that those spots that are not visible have no impact on the calculation. Semantic segmentation networks and 3D universal networks are trained separately. Although the end-to-end feature reduces the aesthetics of the algorithm, it reduces the requirement for 3D annotation.

**3. Data Set**

We took the Apollo data set based on the Unity game engine and rendered the images with a variety of scene structures and visual appearances. The final data set was compiled from three different global maps, each of which featured a unique type of topographic information: highways, urban areas, and residential areas. All of the maps are based on actual locations in Silicon Valley, which can be found in the United States, and each map features lane lines, center lines, and dividing lines that incorporate sufficient ground height variations and turns, as indicated in Table 1

**Table 1.** Examples of composite data.

| | | balanced scenes | | | rarely observed | | | visual variations | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | w/o | w/ | gain | w/o | w/ | gain | w/o | w/ | gain |
| 3D-LaneNet | F-score | 86.4 | 90.0 | +3.6 | 72.0 | 80.9 | +8.9 | 72.5 | 82.7 | +10.5 |
| | AP | 89.3 | 92.0 | +2.7 | 74.6 | 82.0 | +7.4 | 74.9 | 84.8 | +9.9 |
| 3D-GeoNet | F-score | 88.5 | 91.8 | +3.3 | 75.4 | 84.7 | +9.3 | 838 | 90.2 | +6.4 |
| | AP | 91.3 | 93.8 | +2.5 | 79.0 | 86.6 | +7.6 | 86.3 | 92.3 | +6.0 |
| Gen-LaneNet | F-score | 85.1 | 88.1 | +3.0 | 70.0 | 78.0 | +8.0 | 80.9 | 85.3 | +4.4 |
| | AP | 87.6 | 90.1 | +2.5 | 73.0 | 79.0 | +6.0 | 83.8 | 87.2 | +3.4 |

We use detection methods from left to right to create images from highway maps, city maps, and residential maps.
The experimental results are as follows:

**Figure 5.** The original image of the road.

**Figure 6.** Lane detection result 1.

**Figure 7.** Top view 1 of lane detection result.

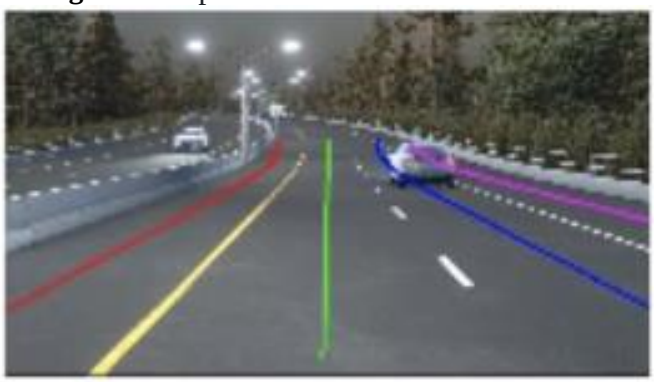

**Figure 8.** Lane detection result 2.

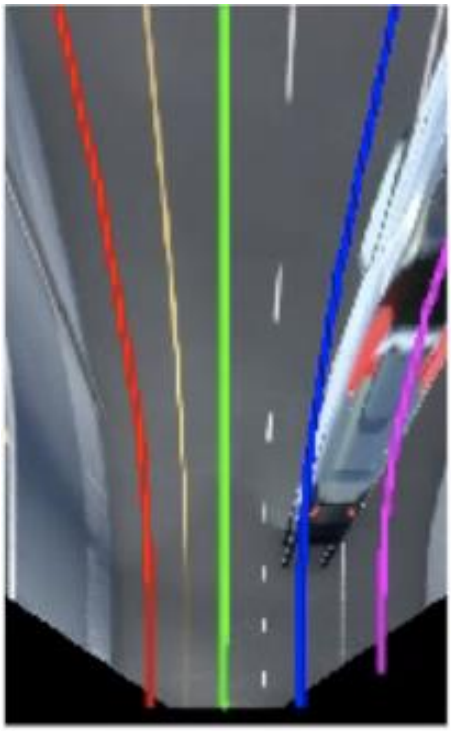

**Figure 9.** Top view 2 of lane detection result.

Conclusion

Lane line recognition can be implemented based on image processing and computer vision algorithms. Commonly used algorithms include edge detection, color segmentation, Hough transform, etc.

The accuracy of lane line recognition is affected by many factors, including ambient lighting conditions, weather conditions, pavement quality, and so on. Complex environmental conditions can lead to identification errors or failures.

Lane line recognition can be used for autonomous driving and assisted driving functions of vehicles. By detecting lane markings in real time, the system can assist vehicles to stay in lanes, change lanes, and perform actions such as turning.

Lane line recognition can also be used for traffic management and road sign recognition. By analyzing lane lines on the road, information such as traffic flow and road congestion can be extracted to assist traffic management decisions.

Lane line recognition technology still faces some challenges, such as multi-lane recognition in complex traffic scenarios, recognition at night or in low-light conditions, and recognition of road signs and construction areas.

In conclusion, the use of computer vision for lane line recognition is a technology with both potential and challenges. With the continuous improvement of algorithms and the improvement of computing power, lane line recognition technology is expected to achieve higher accuracy and robustness in the future, making greater contributions to intelligent transportation and vehicle safety.

## 4. References

[1] Wouter Van Gansbeke; Bert De Brabandere; Davy Neven; Marc Proesmans End to End

Lane Detection through Differential Least Squares Fitting [J] Computer Vision and Pattern Recognition, Volume 1, Issue 2, 2020, P15-17

[2] Pingping Lu, Chen Cui, Shaobing Xu, Huei Peng, Fan Wang SUPER: A New Lane Detection System [J] Computer Vision and Pattern Recognition 2020, Issue 3, Volume 2, P20-21

[3] Ruijin Liu, Zejian Yuan, Tie Liu, Zhiliang Xiong End to End Lane Shape Prediction with Transformers [J] Computer Vision and Pattern Recognition

[4] Zequn Qin, Huanyu Wang, Xi Li Ultra Fast Structure aware Deep Lane Detection [J] Computer Vision and Pattern Recognition Volume 3, Volume 2, 2020, P7-9

[5] DongWu, ManwenLiao, Weitian Zhang, XinggangWang, Xiang Bai, Wenqing Cheng, Wenyu Liu YOLOP: You Only Look Once for Panoptic Driving Perception [J] Computer Vision and Pattern Recognition Volume 2, Issue 1, 2020, P25-27

[6] Ruochen Yin, Biao Yu, Huapeng Wu, Yutao Song, Runxin Niu Fusion Lane: Multi_Sensor Fusion for Lane Marking Semantic Segmentation Using Deep Neural Networks[J] Computer Vision and Pattern Recognition
Volume 3, Issue 3, 2020, P10-P12

[7] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, Zhenguo Li Curve Lane NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending[J]Computer Vision and Pattern Recognition
Volume 2, Issue 2, 2020, P9-11

[8] Netalee Efrat, Max Bluvstein, Noa Garnett, Dan Levi, Shaul Oron, Bat El Shlomo Gen-Lane Net. A Generalized and Scalable Approach for 3D Lane Detection [J] Computer Vision and Pattern Recognition
Volume 1, Issue 1, 2019, P15-P17