

Lane Detection and Classification for Forward Collision Warning System Based on Stereo Vision

Wenjie Song[✉], Yi Yang[✉], Member, IEEE, Mengyin Fu, Yujun Li, and Meiling Wang[✉]

Abstract—This paper presents a lightweight stereo vision-based driving lane detection and classification system to achieve the ego-car’s lateral positioning and forward collision warning to aid advanced driver assistance systems (ADAS). For lane detection, we design a self-adaptive traffic lanes model in Hough Space with a maximum likelihood angle and dynamic pole detection region of interests (ROIs), which is robust to road bumpiness, lane structure changing while the ego-car’s driving and interferential markings on the ground. What’s more, this model can be improved with geographic information system or electronic map to achieve more accurate results. Besides, the 3-D information acquired by stereo matching is used to generate an obstacle mask to reduce irrelevant objects’ interfere and detect forward collision distance. For lane classification, a convolutional neural network is trained by using manually labeled ROI from KITTI data set to classify the left/right-side line of host lane so that we can provide significant information for lane changing strategy making in ADAS. Quantitative experimental evaluation shows good true positive rate on lane detection and classification with a real-time (15Hz) working speed. Experimental results also demonstrate a certain level of system robustness on variation of the environment.

Index Terms—Lane detection and classification, forward collision warning, intelligent car, convolutional neural network, stereo vision.

I. INTRODUCTION

IN RECENT years, Autonomous Driving and ADAS are becoming more and more important to reducing traffic accidents. As one key technology for the intelligent car or Intelligent Transportation System (ITS), driving lane detection and

Manuscript received April 2, 2018; accepted April 27, 2018. Date of publication May 2, 2018; date of current version May 22, 2018. This work was supported in part by the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT-16R06 and Grant T2014224, in part by the National Natural Science Foundation of China under Grant NSFC 61105092, Grant 61173076, Grant 61473042, and Grant 91120003, in part by the BNSF under Grant 4132042, and in part by the Beijing higher education young elite teacher project under Grant YETP1215. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ioannis Raptis. (*Corresponding author: Yi Yang.*)

W. Song is with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China, and also with Princeton University, Princeton, NJ 08544, USA.

Y. Yang and M. Wang are with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China (e-mail: yang_yi@bit.edu.cn).

M. Fu is with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China, and also with the Nanjing University of Science and Technology, Nanjing 210094, China.

Y. Li is with Princeton University, Princeton, NJ 08544, USA, and also with Shanghai Jiao Tong University, Shanghai 200240, China.

Digital Object Identifier 10.1109/JSEN.2018.2832291

classification is not only able to solve the problem of accurate vehicle lateral position estimation, but also provide more traffic information for driving ego-car to make better decision. Therefore, it has attracted a widespread attention from plenty of institutes and automobile technology companies. For example, Mobileye and Bosch are developing some low-cost but effective products with Forward Collision Warning (FCW) or Lane Departure Warning (LDW) function by combining camera with mmw-radar for autonomous driving in structured environments [1]. Many automobile companies, such as Benz, BMW and Tesla, are also adding similar functions to their car through cooperating with other famous institutes, such as KIT Autonomous Vision Group and Daimler Research Group.

Considering that stereo camera is desirable in autonomous driving to gain rich information of the surrounding area for safe navigation [2]–[4], we developed a stereo vision-based lane detection and classification method for FCW system to obtain the accurate drivable area and detect forward collision distance in structured environments. The main idea includes three parts: First, process stereo images to conduct obstacle segmentation and fuse the segmentation result with a self-adaptive traffic lanes model to detect parallel lanes; Second, select host lane from the detected lanes and classify its each side to be dashed, solid, dashed-solid, solid-dashed, double-solid, curb; Third, estimate collision distance in host lane and potential collision distance in its adjacent lane which the ego-car can change into without violating traffic rules. As shown in Fig. 1, red line denotes the side line of host lane, light-blue line denotes the side line of other lanes, types of each side line of host lane are shown at the top-left/top-right corner, dark-blue area denotes the nearest obstacle detected in host lane, yellow area denotes the nearest obstacle in the enterable adjacent lane. What needs to be explained is that there is no yellow obstacle detected in the third image (in Fig. 1) because the left-side line of the host lane is classified to be solid, which can not be passed through. Thus, the developed method can provide valuable information for forward collision warning or lane changing strategy making.

The main contributions of this paper include: We employ a stereo camera to generate an obstacle mask in the bird-view image with few malposition even though the road is bumpy so that most interfere from irrelevant objects can be eliminated; We design a self-adaptive traffic lanes model in Hough Space with a maximum likelihood angle ROI and dynamic poles detection ROIs, which is robust to road bumpiness, lane structure changing while the ego-car’s driving and interferential

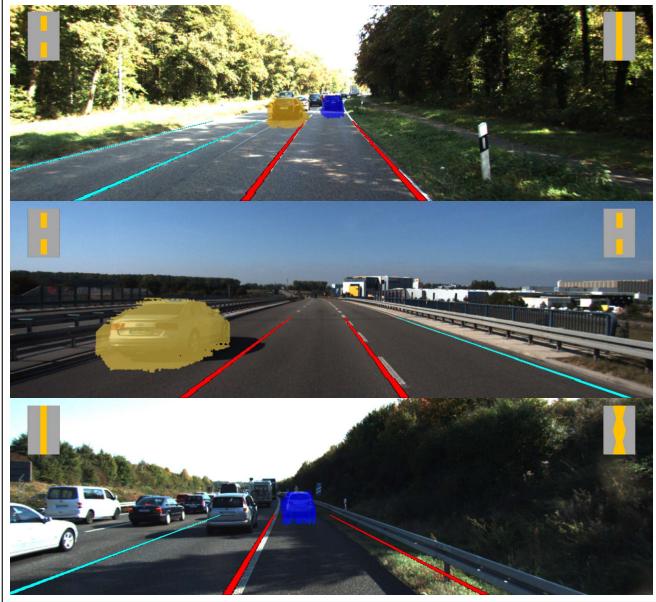


Fig. 1. Lane detection and classification for forward collision warning system based on stereo vision.

markings on the ground. What's more, the proposed traffic lanes model can be improved with geographic information system (GIS) or electronic map to achieve more accurate lane detection. We use a CNN for lane markings classification and achieve high accuracy of classification; We propose such a lightweight system that can provide lateral positioning information and forward (potential) collision distance to the ego-car only by using a stereo camera; The presented system should be effective in structured environments, especially the highway environment.

II. RELATED WORK

There is no doubt that vision-based collision warning is an important component of ADAS or Autonomous Driving. What's more, lane analysis in a collision warning system is the indispensable core technology for autonomous or aided driving in structured environments. Therefore, the generic problem of lane detection and recognition has been studied extensively by computer vision or robotic community. Some survey papers, such as [5], [6], and [7], have proposed and compared a wide variety of lane detection techniques with different traffic lanes models that were raised in the past few years. Based on the result of lane detection and classification, researchers employ different methods, such as vehicle detection, obstacle detection and so on, to develop different FCW or lane changing assist (LCA) system. In this section, we mainly discuss these approaches from two aspects: lane detection and classification, lane analysis-based FCW or LCA system.

For the first aspect, that is lane detection and classification, plenty of excellent and representative methods were proposed and applied to different platforms in recent years. Some researchers prefer to detect lane in the original image without time or space transformation, which has been studied in [8]–[14]. Among them, [8] and [9] mainly focused on

multiple lanes detection based on the vanishing points. This kind of methods were built based on a same constraint condition, that is parallel lines on the ground converge to a vanishing point in the image coordinate system. Some works, like [10] and [11], were based on Random Sample Consensus (RANSAC). For example, [11] converted the original RGB image into a “log-chromaticity” space to conduct road segmentation, used a low level image feature called “ridge” to detect lane marking and utilized RANSAC to estimate the road geometry. Besides, some other algorithms were also applied to lane detection in the original image, such as Hough Transform [13], B-Snake [14] and so on. Considering that it may become easier for lane detection after time or space transformation, more methods work in other space or coordinate system. Among them, some researchers, like [15]–[17], tried to generate time-slice (TS) or spatiotemporal image for lane detection through stacking rows from the original image. For example, [16] aligned scan lines along time axis and applied Hough Transform to detect lane points. Other researchers detected lanes in a bird-view image by conducting Inverse Perspective Mapping (IPM) for the original image, which has been studied in [18]–[21]. This kind of methods are built in the vehicle coordinate system, in which lanes on the ground are approximatively parallel to each other. For example, [19] employed K-Means to cluster lines and fitted the lane discontinuous based on a lanes model. We also conducted similar work in our previous research [22]. In which, we used a common Hough Transform-based lane detection algorithm in a panoramic bird-view image that was transferred from the image captured by Ladybug camera. In addition to these traditional methods, more and more researchers turned to deep learning-based lane detection, which has been studied in [23]–[26]. For example, [24] developed two CNNs in their system, one multitask CNN was used for detecting the presence of the road marking and its geometric attributes, the other was used for structural prediction. Reference [25] presented a Dual-View CNN for lane detection and stated their method was more accurate and robust than the state-of-the-art. Moreover, some researchers also fused other sensors with camera to develop lane detection system. For example, [27] fused lidar with a monocular camera to build a novel real-time optimal-drivable-region and lane detection system. And our previous work [22] also employed a 3D lidar to improve the system reliability. However, we found the mask created by lidar was always accurate in flat road environments, but not good on the bumpy road because the IPM image was always twisty in these environments. In other words, lidar mask was malposed when projected to the twisty IPM image because the road is bumpy. Besides, lidar (especially 3D lidar) is always much more expensive than stereo camera. Therefore, to decrease the malposition, we employed a stereo system instead of lidar to create the obstacle mask. Meanwhile, we also established a more robust self-adaptive traffic lanes model in Hough Space to solve the road bumpiness. In this way, the system cost can be lowered greatly and more perceptual information can be obtained.

Compared with lane detection, lane classification is equally important to ADAS or Autonomous Driving. However, until

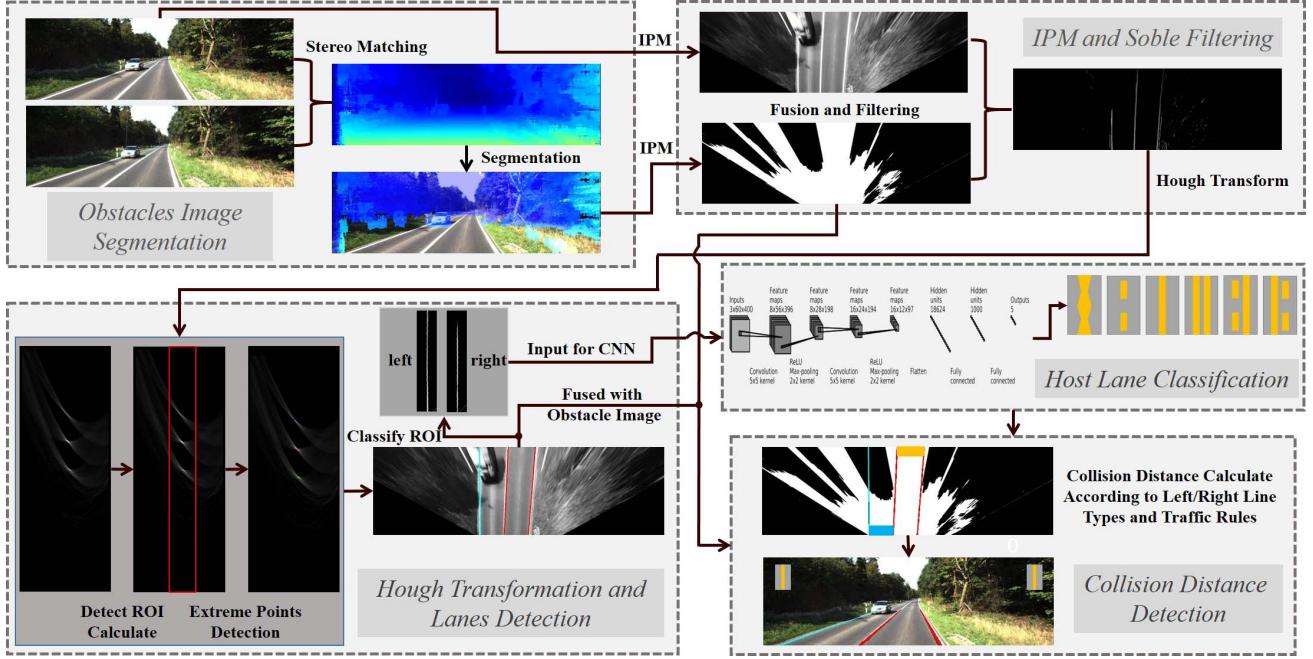


Fig. 2. Software architecture of lane detection and classification in the forward collision warning system.

now, far less effort has been devoted to this kind of problem. Reference [28] developed a Bayesian classifier based on mixtures of Gaussians to classify lane markings into five categories. Their results proved that their method can realize real-time lane detection and classification in some simple environments. Reference [29] presented a LCA system in their paper, in which they classified host lane into two types (dashed or solid) to decide whether lane changing is possible or not. However, these two methods focused on the whole image, which may lead to high computational cost and low recognition rate. Inspired by them, we employed a CNN to recognize the picked host lane ROI (classify it into dashed, solid, dashed-solid, solid-dashed, double-solid or curb) so that forward collision warning and lane changing assist can be realized.

For the second aspect, real-time obstacles segmentation or drivable area detection is necessary for a lane analysis-based FCW or LCA system. Some researchers utilized range sensors (lidar or radar) to detect the front obstacle or vehicle in their system. For example, [27] used a 2D SICK lidar to find obstacles in each detected lane. But more and more researchers prefer to employ cameras for obstacle or vehicle detection in a FCW or LCA system. For example, [30]–[32] developed a FCW or LCA system by using one monocular camera to realize lane detection and vehicles tracking. References [33] and [34] employed a stereo camera to get more information for FCW or LCA system designing. For the stereo vision-based system, disparity or depth image was always first calculated through stereo matching for obstacles segmentation, which has been studied in [35]–[38] and [39]. In our system, we only use a stereo camera to achieve lane detection and classification in structured environments. On the one hand, obstacles detection result is used to remove noise or distractions on the road (cars, trees, etc.) for lane detection. On the other hand, lane detection

result is utilized to obtain more accurate traversable region in the specified host lane or enterable adjacent lanes.

III. SYSTEM OVERVIEW

The whole system architecture, as shown in Fig. 2, mainly includes five modules: Obstacles Image Segmentation, IPM and Sobel Filtering, Hough Transform and Lane Detection, Host Lane Classification, Collision Distance Detection. In the first module, we combine an efficient GPU-based stereo matching algorithm developed by Daniel [2] with an UV-disparity segmentation algorithm which is slightly modified from the traditional UV-disparity algorithm [40] to get obstacles image. In the second module, the original image and obstacles image are projected to the ground through IPM that was developed by Aly [18]. Next, we utilize a Sobel filter that was ever used in our previous work [22] to fuse these two bird-view images together to get a low-noise gray bird-view map. In the third module, we firstly conduct Hough Transform [41] for the low-noise bird-view map to get a voting map in Hough Space. Then, the maximum likelihood angle range is obtained as the detect ROI through the angular-axis histogram statistic in Hough Space. Next, we find the maximum point (basic pole) in the detect ROI as the maximum likelihood lane. Then, we obtain all the other extreme points or poles as parallel lanes based on a designed traffic lanes model with dynamic pole detection ROIs. What's more, the two detected lines (on the left/right of the ego-car) that are closest to the ego-car (middle point of the bottom in the bird-view image) are selected as host lane. If only one side line is acquired, the other side line of host lane can be supplemented according to the traffic lanes model. Once host lane is gotten, we can get the left/right classify ROIs in the low-noise gray bird-view map for host lane classification. In the fourth module, these two selected classify

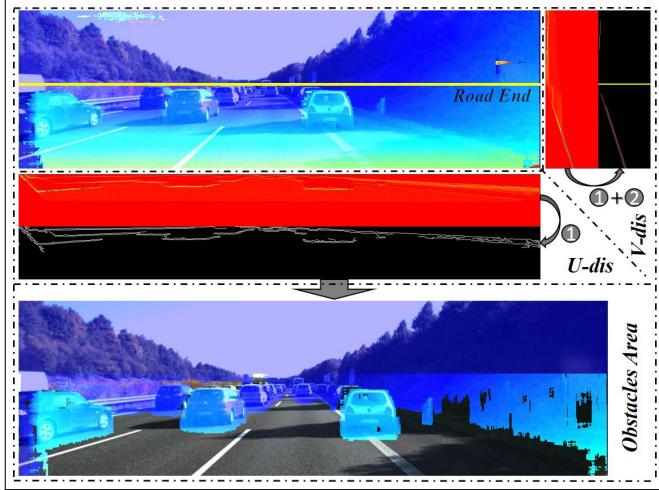


Fig. 3. Obstacles image segmentation through the modified UV-disparity algorithm.

ROIs are imported into a trained CNN which is improved based on LeNet-5 [42] to guess their categories (five outputs in CNN). What needs to be explained is that if the side line of host lane is supplemented (not detected in Hough Space), its type will be set to curb. In the fifth module, we design several detection rectangles to calculate the collision distance in host lane and its adjacent lanes based on the host lane's type and traffic rules. For example in Fig. 2, collision distance in the adjacent lane is set to 0 because the left-side line is solid even though the front car is quite far away from the ego-car. These five modules work together harmoniously can make the system more robust and efficient in structured environments. The whole system has been tested to get good results by using the public dataset KITTI [3]. Besides, we also tested it on our own platforms in different environments to prove its effectiveness.

IV. LANE DETECTION AND CLASSIFICATION FOR COLLISION WARNING

In this section, these five modules in our system are introduced according to the system architecture in Fig. 2, respectively. Besides, experimental results of each module are also presented.

A. Obstacles Image Segmentation

In this module, the traditional UV-disparity algorithm is improved to conduct obstacles segmentation based on the real-time dense disparity image, which is acquired through a GPU-based Semi-Global Matching (SGM) algorithm from [2]. The improved segmentation method (Fig. 3) has been introduced in detail in our previous work [43]. First, we get V-disparity by processing the original disparity image. Then, V-disparity information is used for calculating the road end to split the image into two areas: the whole upper area is regarded as obstacles and the area below the road end is selected to calculate U-disparity for accurate obstacles area detection. Specifically, V-disparity process includes two steps. The first

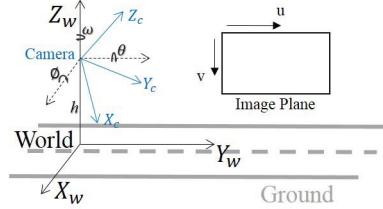


Fig. 4. IPM coordinates (world, camera, and image frames).

step is edge detection through Canny algorithm. The second step includes detecting lines in the obtained edge image, calculating their intercepts on the y-axis, and selecting the true road end (yellow line in Fig. 3) among all the intercepts. However, U-disparity process only employs the first step to get edge image. Then, the white pixels in this edge image are projected to the original image for obstacle pixels detection. In [43], we have presented segmentation results in different environments, such as night, rains, snows, strong light or other bad weather conditions and the hilly terrain, to prove the robustness of this segmentation method.

B. IPM and Sobel Filtering

To revert the actual shape and relative position of lane markings on the road, a bird-view image is generated through IPM in our system. A real-time and robust IPM algorithm in [18] are employed for this process. To perform the transformation, some extrinsic parameters, including camera intrinsic (focal length and optical center), height above ground h , pitch angle θ , yaw angle ω , roll angle ϕ , need to be measured. Thus, the bird-view image with specified unit can be obtained through transform between the IPM coordinates. As shown in Fig. 4, (X_w, Y_w, Z_w) is the world coordinate, (X_c, Y_c, Z_c) is the camera coordinate and (u, v) is the image plane. In our system, only the area below the road end line is selected as ROI to be projected onto the ground plane for the bird-view images. For the original left image, its gray image can be gotten through formula (1) to enhance the lane feature, and its bird-view image is obtained through IPM as shown in Fig. 5(a). Meanwhile, we can also get the bird-view image of the obstacles image as a mask (Fig. 5(b)) in the vehicle coordinate system, in which white area is the traversable area.

$$\text{Gray} = \text{Red} * 0.8 + \text{Green} * 0.15 + \text{Blue} * 0.05 \quad (1)$$

$$G_{i,j} = \exp\left(-\frac{j^2}{4\delta^2}\right)\left[\exp\left(-\frac{i^2}{2\delta^2}\right) - \exp\left(-\frac{i^2}{64\delta^2}\right)\right] \quad (2)$$

Then, a two dimensional Gaussian kernel is designed according to formula (2) for filtering the gray bird-view image to extract vertical lane markings. And the parameter δ is adjusted according to the expected width of lanes. Besides, only $q\%$ brightest part of the filtered image is selected to get the threshold image (Fig. 6) through histogram calculation because lane markings have much higher value than others. In detail, considering that plenty of noise has been removed and vertical lane marking feature has been greatly heightened after filtering, the selection of threshold q doesn't have to

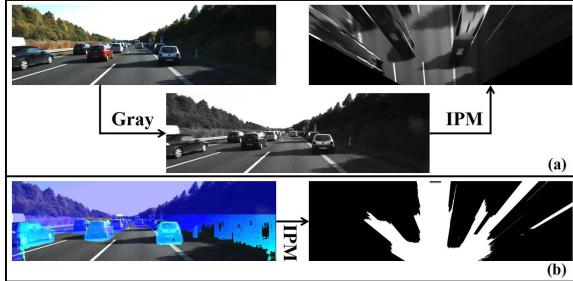


Fig. 5. IPM for the original image and the obstacles image.

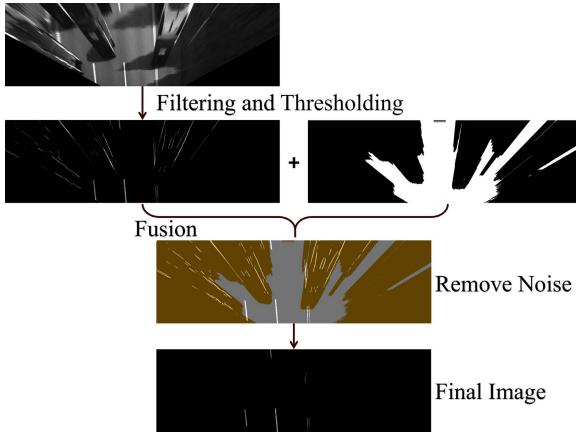


Fig. 6. Remove noise through Sobel filter and fusion with obstacle mask.

be very strict. In practice, we set q to be 5 in our system after testing with KITTI dataset and our own dataset collected in realist scenarios. Next, we employ the obstacles image to create a mask on the threshold image as shown in Fig. 6 to remove the noise caused by other objects (cars, tree, etc.) on the road. It is obvious that a mass of noise is removed successfully after fusing these two images, which makes lane detection much easier.

To adapt to the real driving situation, we test the performance of this module in different environments. In general, researchers usually conduct IPM on the premise that the road is deemed to be perfectly planer. However, the actual road in real life is not always planar, even quite bumpy sometimes. Therefore, it is full of challenge for IPM process. In other words, it is so hard to get the ideal bird-view images when the ego-car drives on the bumpy road. For this reason, we specially test this module on our own platform in different road conditions including planar, uphill, valley, downhill and crest around Princeton University, where the road condition changes frequently. As shown in Fig. 7, we select a ROI (red area) as the original image in the first row. The second row is the obstacles segmentation image. The third row shows the bird-view image and the fourth row shows the low-noise bird-view image. Specifically, Fig. 7(a) (the first column) is gotten on the planar road, Fig. 7(b) is uphill, Fig. 7(c) is valley, Fig. 7(d) is downhill, Fig. 7(e) is crest. From the result in Fig. 7, we can find that obstacles segmentation and noise removing process are only affected slightly by the bumpiness

so that we can get a good low-noise bird-view gray image. However, it is obvious that IPM process is affected seriously by the bumpiness. For example in Fig. 7, the bird-view image on the planar road is much better than others. Especially when the ego-car is near to the valley (Fig. 7c), the bird-view image becomes quite twisty (parallel lines in this image are intersectant). To realize robust lane detection in such environments, the traffic lanes model must be able to work normally based on these results, which is introduced in the next sub-section.

C. Hough Transformation and Lane Detection

As we introduce in Section III, a self-adaptive traffic lanes model is designed in Hough Space to detect all the parallel lines. To get the $r - \theta$ space (Hough Space), Hough Transform is conducted for the low-noise bird-view image as shown in Fig. 8 (from (a) to (b1)). In the $r - \theta$ space, an extreme point (pole) indicates a line in the real world. However, as shown in Fig. 7, parallel lines on the ground are not always paralleled in the bird-view image because IPM process may be unfaithful on the bumpy road. Thus, poles that indicate the parallel lines don't always share the same angle value in the $r - \theta$ space. Therefore, in order to obtain an accurate pole for each parallel line, we design a set of dynamic ROIs based on a self-adaptive traffic lanes model.

First, we get the maximum likelihood angle through angle statistics to obtain an angle ROI instead of directly detecting poles in the whole space, as shown in Fig. 8(b2). Specifically, the score of each angle is calculated according to formula (3) to get a yellow curve as presented in Fig. 8 and the max angle point among them is selected as the maximum likelihood angle. Then, angle range W (set to be 16° in our system) is set as ROI width to get the angle ROI for poles detection. In formula (3), $A(i)$ is the i^{th} angle score, $rows$ is the row number of $r - \theta$ space, w is a filter width ($w = W \cdot cols / 180$), $cols$ is the column number of $r - \theta$ space, $g(i + k, j)$ is equal to the gray value of point $(i + k, j)$ in $r - \theta$ space when the gray value is greater than a threshold T (T can judge whether a point is possible to be a candidate), otherwise, $g(i + k, j) = 0$. Finally, the maximum point in the angle ROI (Max Point in Fig. 8) is detected as the basic pole.

$$A(i) = \sum_{j=0}^{j=rows} \left\{ \sum_{k=-\frac{w}{2}}^{k=\frac{w}{2}} \left[\left(\frac{w}{2} + 1 - abs(k) \right) \cdot g(i+k, j) \right] \right\} \quad (3)$$

In the second step (Fig. 8(b3)), distance (Lane Range) between the adjacent lines is calculated as an important component of the traffic lanes model. To initialize, Lane Range is set to be $4m$ in our system. Next, if the gray value of Max Point is greater than T , it can be regarded as a line. Then, two new pole detection ROIs are generated based on it. Center of the up ROI (X_{roi}^u, Y_{roi}^u) and center of the down ROI (X_{roi}^d, Y_{roi}^d) are calculated according to Max Point (X_{max}, Y_{max}), as shown in formula (4). Besides, according to common sense and testing, we choose a suitable size for all the pole detection ROIs (height is set to be $0.5m$ and width is set to be W in our system). Then, poles in these two ROIs

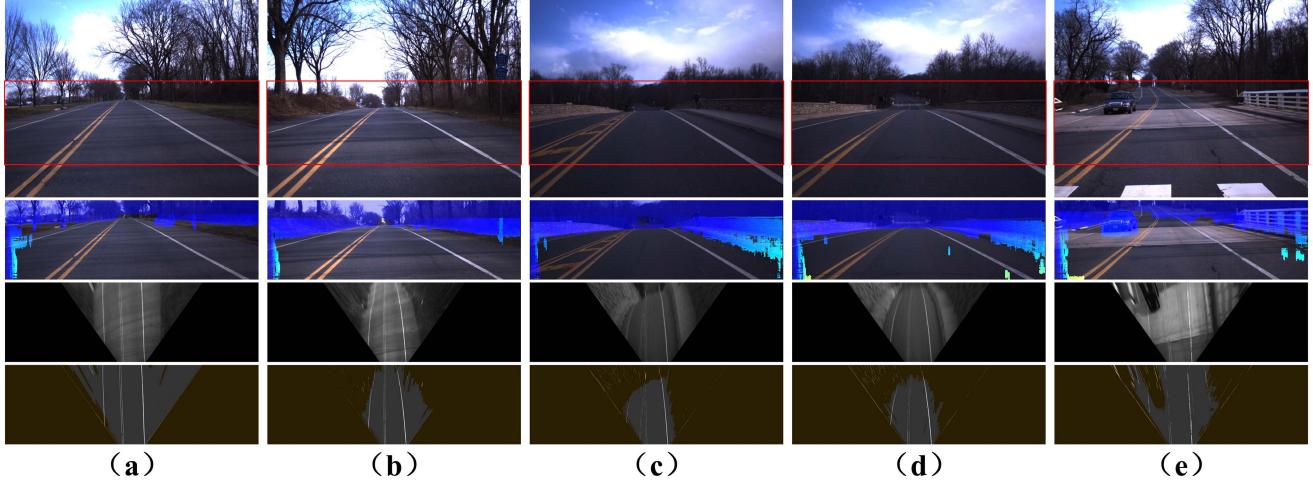


Fig. 7. Results of IPM process and obstacles fusion in different road conditions: (a) planar; (b) uphill; (c) valley; (d) downhill; and (e) crest.

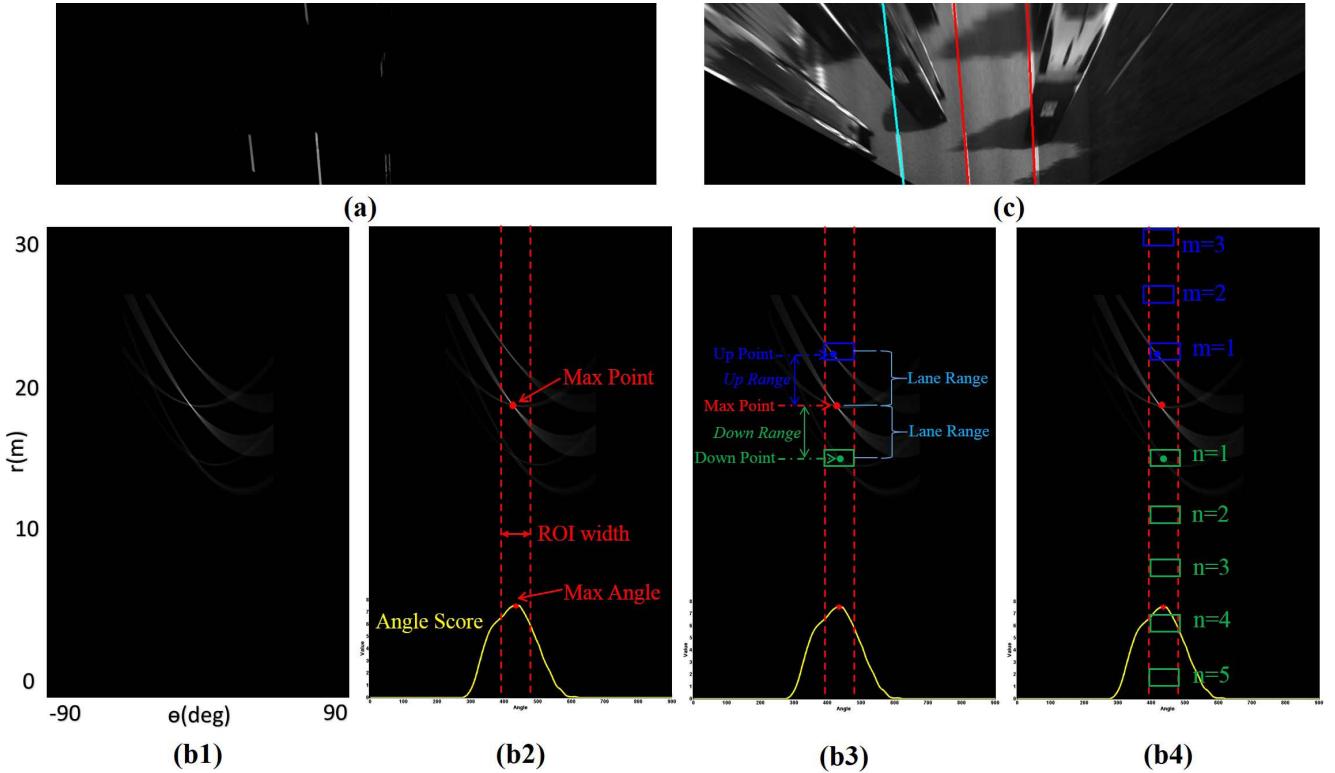


Fig. 8. Hough Transform and the traffic lane model for lane detection.

are detected as Up Point and Down Point, respectively. And the range difference between Up Point and Max Point (Down Point and Max Point) is calculated as Up Range (Down Range). In our system, when the gray value $g^u(g^d)$ of Up Point (Down Point) is greater than $0.2g_{max}$ (g_{max} is the gray value of Max Point), it can be regarded as a valid pole. Thus, when both Up Point and Down Point are valid, Lane Range will be set to be Up Range if $g^u > g^d$ (or Down Range if $g^u \leq g^d$). When only one of them is valid, Lane Range is set to be the Range of the valid pole. When neither of them is valid, Lane Range keeps unchanged. What's more, the parameters including Lane Range, number of lanes and so on also can be acquired in advance if we can get access to GIS or electronic

map information. If so, the model-based lane detection would be more accurate.

$$\begin{cases} X_{roi}^u = X_{max} & Y_{roi}^u = Y_{max} - \text{LaneRange} \\ X_{roi}^d = X_{max} & Y_{roi}^d = Y_{max} + \text{LaneRange} \end{cases} \quad (4)$$

In the third step (Fig. 8(b4)), all the other pole detection ROIs are obtained dynamically for multiple lines detection based on their superior detected pole coordinate and the current lane model. Specifically, for a up ROI, its superior detected pole is located in its adjacent ROI under it. For a down ROI, its superior detected pole is located in its adjacent ROI above it. In this way, the coordinate of the next ROI is calculated according to formula (5) and its size is same as that

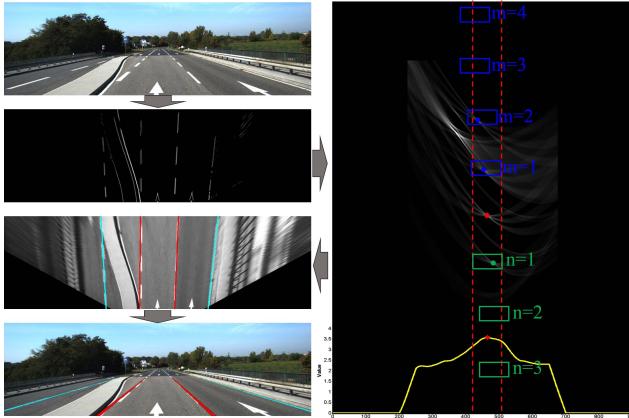


Fig. 9. Lane detection model is robust for the disturbance caused by other types of markings.

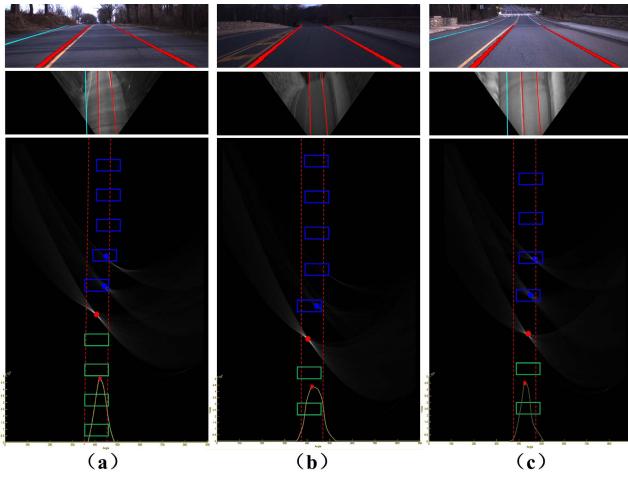


Fig. 10. Lane detection model is robust for the bumpy environments, such as: (a) uphill, (b) valley, and (c) downhill.

in the second step. In formula (5), $(X_{roi}^u(m), Y_{roi}^u(m))$ is the center coordinate of the m^{th} up ROI, $(X_{max}^u(m-1), Y_{max}^u(m-1))$ is the coordinate of the detected pole in the $(m-1)^{th}$ ROI, $(X_{roi}^d(n), Y_{roi}^d(n))$ is the center coordinate of the n^{th} down ROI, $(X_{max}^d(n-1), Y_{max}^d(n-1))$ is the coordinate of the detected pole in the $(n-1)^{th}$ ROI. What needs to be explained is that, when the gray value of the detected pole in one ROI is not greater than $0.2g_{max}$, the pole will be regarded as an invalid pole. Then, it should be set to be center of this ROI, namely $(X_{max}^d(n-1), Y_{max}^d(n-1)) = (X_{roi}^d(n-1), Y_{roi}^d(n-1))$, $(X_{max}^u(m-1), Y_{max}^u(m-1)) = (X_{roi}^u(m-1), Y_{roi}^u(m-1))$. Thus, we can get all the valid poles and ROIs in $r-\theta$ space. For example in Fig. 8(b4), the first up ROI ($m=1$) and the first down ROI ($n=1$) are valid ROIs, their poles are valid poles. In this way, three lines (three valid poles, including the basic pole) are detected and drawn in the bird-view image, as shown in Fig. 8(b4) and Fig. 8(c).

$$\begin{cases} X_{roi}^u(m) = X_{max}^u(m-1) \\ Y_{roi}^u(m) = Y_{max}^u(m-1) - \text{LaneRange} \\ X_{roi}^d(n) = X_{max}^d(n-1) \\ Y_{roi}^d(n) = Y_{max}^d(n-1) + \text{LaneRange} \end{cases} \quad (5)$$

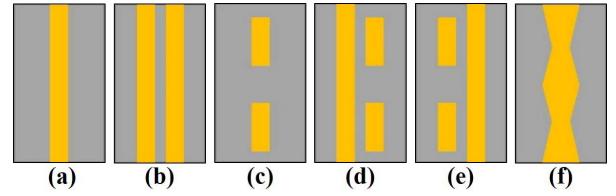


Fig. 11. Types of the side line of host lane: (a) single-solid; (b) double-solid; (c) dashed; (d) solid-dashed; (e) dashed-solid; and (f) curb.

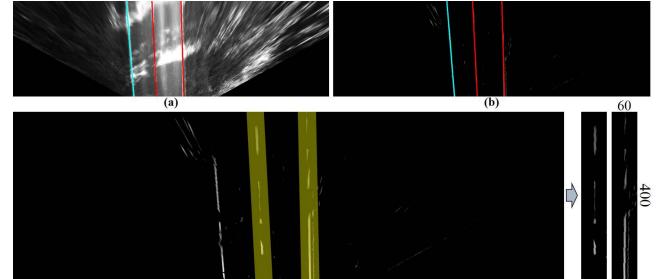


Fig. 12. Selecting classify ROI images based on the lane detection result as the input of CNN.

Through testing this module in different environments, we find that the traffic lanes model in Hough Space with the angle ROI selection and dynamic poles detection ROIs indeed makes our system much more robust. On the one hand, the angle ROI can effectively reduce the poles searching range so that other interferential information is ignored. On the other hand, the dynamic poles detection ROIs can overcome the distortion of the bird-view image and exclude plenty of noise (for example, arrows between lines) on the road. For example, when there are many other types of interferential road markings as shown in Fig. 9, our model can successfully detect the parallel straight lines while ignoring the oblique lines and the arrows although they own very high voting values in Hough Space. Besides, as shown in Fig. 10, the designed model also can effectively detect the traffic lanes when the IPM process can not get a good bird-view image (sometimes quite twisty) because of road bumpiness. In Fig. 10, lanes on the uphill, valley, downhill road can be detected and projected back into the image plane accurately.

D. Host Lane Classification

Besides lane detection and localization, we also develop a method for lane classification. In order to simplify the problem, we assume that our system only works in some special driving environments, such as the expressway, state road and so on. In these situations, the type of host lane is vital for a driving ego-car. In our system, we select two closest lines to the ego-car (center point of u-axle in bird-view image) as the host lane among all the left-side and right-side lines, respectively. However, if only one side line is detected, the other side line will be supplemented according to Lane Range to obtain the host lane. Thus, host lane can be selected as the two red lines shown in Fig. 8(c). The basic idea of this module includes selecting classify ROIs for the left/right-side line and importing these two ROIs into a trained CNN

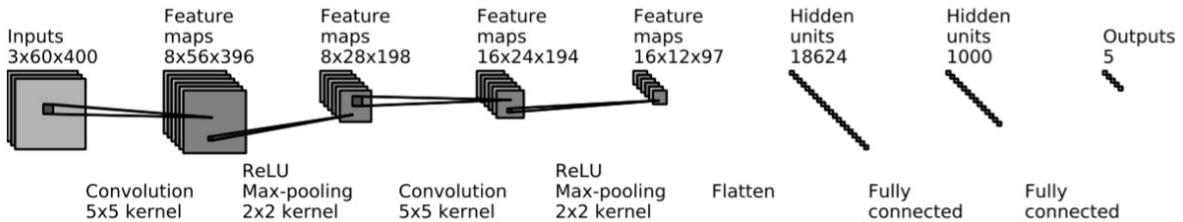


Fig. 13. Structure of the Convolutional Neural Network employed in our system.

to conduct prediction, respectively. If the side line is supplementary (not detected in Hough Space), it will be classified to be curb. Otherwise, it will be classified to be five possible types through CNN: single-solid, double-solid, dashed, solid-dashed and dashed-solid. These six types are shown in Fig. 11. In our system, we select these two ROIs (60×400) from the low-noise bird-view image (1280×400) as shown in Fig. 12. What needs to be noticed is that the ROI width must be suitable to cover the detected line even when IPM process is inaccurate because of road bumpiness. Specifically, for the planar road, the ROI width (60 pixels in the bird-view image) indicates around 1.2m. In detail, because the detected lines are almost not upright (always oblique), the selected parallelogram ROIs (yellow area in Fig. 12) in the bird-view image is transformed to rectangle through $g(i, j) = G(I - \frac{w}{2} + i, J)$, where, $g(i, j)$ is the gray value in the desired rectangle ROI image, G is the gray value in the bird-view image, (I, J) is the coordinate of the point on the detected line (in each row of the bird-view image, there is only one point on the line), w is the ROI width ($w = 60$). In this way, we can get these two rectangle ROIs (two 60×400 images, as shown in Fig. 12) as the inputs for CNN. CNN has shown its outstanding ability in object recognition and natural image classification in the famous ImageNet competition. In order to run the whole system in real time, we focus on the shallow networks, which require much less computation time than deep networks, such as VGGNet [44] (a widely used deep convolutional network developed and trained by Oxford's renowned Visual Geometry Group (VGG), achieving very good performance on the ImageNet dataset). As shown in Fig. 13, the employed CNN structure is used from Caffe [45], which is the deep learning framework developed at the Berkeley Vision and Learning Center. The network is fed with the two generated 60×400 grayscale images, consisting of 2 convolution, 2 max pooling and 2 fully connected layers. Each convolution network is followed by a rectified linear unit (ReLU). In practice, we create the train data and the test data by ourselves. We select enough classify ROI images by using both KITTI dataset and our own dataset, set correct label for them manually and convert them into the available format based on the Caffe framework. The dataset with 22000 images is randomly divided to 16000 images as training dataset and 6000 images as test dataset. Each image is labeled to one of five types (except curb) as shown in Fig. 14. We use cross entropy to measure the final result and set the batch size as 32. The base learning rate is set to be 0.001 at the origin, and

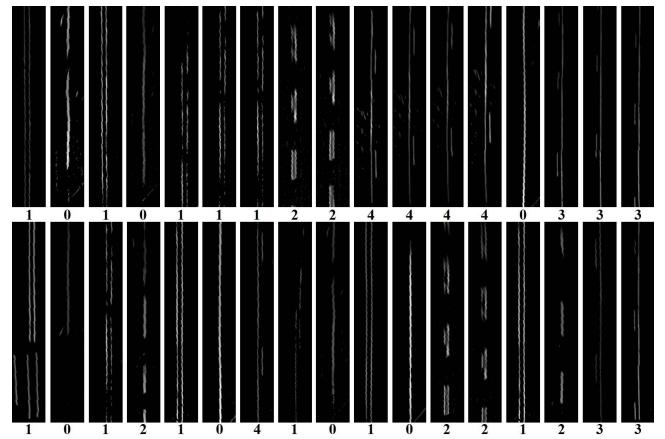


Fig. 14. Samples of the generated train/test data with label. 0: solid. 1: double solid. 2: dashed. 3: dashed-solid. 4: solid-dashed.

multiples 0.999 at each iteration. The momentum is set to 0.9 in the network. The model has been trained for approximately five hours on GTX-1080TI, resulting in accuracy 99.6%.

As we explain above, the inputs (Fig. 14) for CNN are selected from the preprocessed image, which only contain the line features after filtering out the noise through Sobel filtering and fusion. Although we have gotten the perfect test accuracy through the presented method, we also tried other end-to-end classification model by mapping original ROI images to types directly on a shadow network. In other words, we want to test whether the CNN can train the unprocessed ROI images to get a better filter than that we designed. Therefore, we selected classify ROIs from the original bird-view image instead of the final low-noise bird-view image, as shown in Fig. 15. Some of the generated train/test images are presented in Fig. 15. However, we found it hard to get a better result even though we changed the CNN structure for many times, trained much more images or used other more complex networks. In fact, the best test accuracy among all the experiments could only achieve about 60% when we used the original ROI images as inputs. So we didn't adopt this idea, but kept our previous method, that is selecting ROI images from the final low-noise bird-view image as inputs for the whole classification system.

E. Collision Distance Detection

Considering that obstacles in host lane and its adjacent lanes are usually more important to the driving ego-car, our system

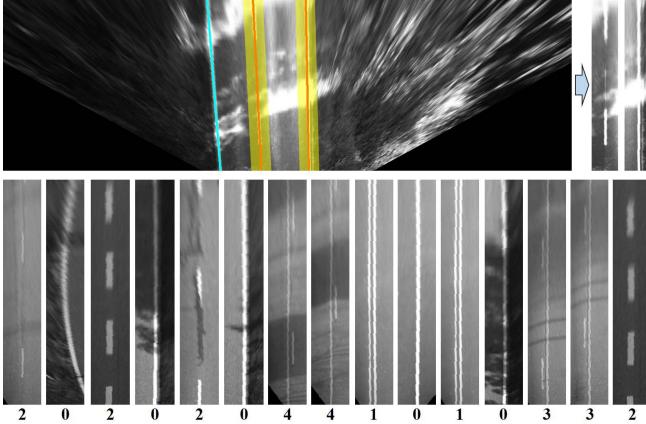


Fig. 15. Selecting ROI images directly from original bird-view image and generating the corresponding train/test data (label is same as Fig. 14).

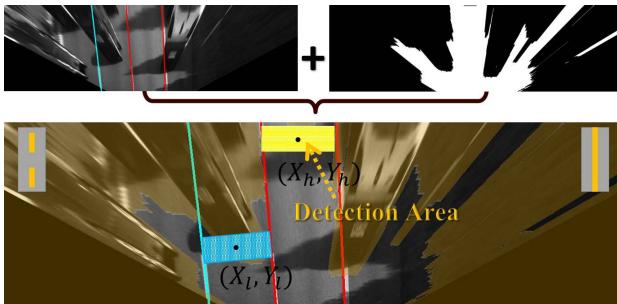


Fig. 16. Collision distance detection in the detected host lane and its passable adjacent lanes.

only focuses on the forward collision distance detection in these lanes. In this module, we employ a simple and coarse collision distance detection method to present the anti-collision concept in our system, which is not the key technology in this paper. Therefore, we spend less efforts in this module. Specifically, when both two side lines are solid (or other types for no crossing), we only detect the forward collision distance in host lane. Otherwise, the adjacent lane with the allowing crossing type (dashed, for example) is also taken into account. For example, in Fig. 16, the yellow polygon is the detection block in host lane and the blue polygon is for the left adjacent lane (no detection block in the right adjacent lane) because the left-side line is dashed and the right-side line is solid. These blocks slide from the bottom to the top in host lane or passable adjacent lanes until obstacles are detected. We have tested the obstacle size for different kinds of objects, such as car, pedestrian, telegraph pole. Their size in the corresponding bird-view image is always large enough. According to the testing results, the block is set to stop sliding if 25% (or more) of it are obstacle pixels. Thus, the forward collision distance in host lane is $(Y_d+offset)$ and the potential forward collision distance in the left adjacent lane is $(Y_l+offset)$. (X_d, Y_d) and (X_l, Y_l) are the center of the sliding blocks and $offset$ is a constant related to IPM process.

V. EXPERIMENTAL RESULTS

For the obstacle mask, we compared the vision-based segmentation mask with the 2D lidar mask in different



Fig. 17. Comparison results between the vision-based segmentation mask (white area) and the 2D lidar mask (green area) on the bumpy road (a) and the flat road (b).

TABLE I
TWO DIFFERENT DATASETS USED FOR THE EVALUATION

	Scenarios	Frames	Target Host	Adjacent	Target Adjacent
KITTI	Road_0004	339	339	321	159
	Road_0015	297	297	297	155
	Road_0027	188	188	188	186
	Road_0028	430	430	430	429
	Road_0042	835	835	1592	915
Ours	Daytime1	724	724	787	710
	Daytime2	1480	1480	1427	1382
	Night1	1004	987	987	785
	Night2	427	427	374	353
	Night3	535	496	458	415

environments, as shown in Fig. 17. It's obvious that the lidar mask is much more accurate than the vision-based segmentation mask on the flat road. However, because IPM process for obstacles segmentation image shares the same parameter with that for left image (Fig. 5), the malposition of the vision-based segmentation mask is much slighter compared with the 2D lidar mask, especially on the bumpy road. In other words, the vision-based segmentation mask can work better than the 2D lidar mask when the bird-view image is twisty.

In order to evaluate the performance of lane detection and classification, experiments in different kinds of environments were also conducted based on two different datasets. As presented in TABLE I, we mainly employed the public dataset KITTI and our own dataset. KITTI dataset, which was recorded from a moving vehicle driving through the city of Karlsruhe, has always been used to evaluate their methods by plenty of researchers. We mainly present test results of the Road Category Raw Data (for example, *Road_0004* represents *2011_09_26_drive_0004* in the Road Category). Besides, considering that KITTI dataset only includes scenarios at daytime on flat roads, we also collected some other challenging data by using our own platform to test the system at night or on bumpy roads. Specifically, we used a stereo camera Bumblebee-xb3, whose baseline was 24cm, to collect 1280 × 960 images and selected suitable ROI (1280 × 400, as shown in Fig. 7) as final images around Princeton University, where the road was always bumpy. What's more, all the images in these two datasets were uniformly resized into 1280 × 400 in our system. TABLE I describes the datasets of these scenarios. We manually counted the number of target host lanes, adjacent lanes and target adjacent lanes in the scenarios. In detail, **Target Host** is the number of host lanes that need to be

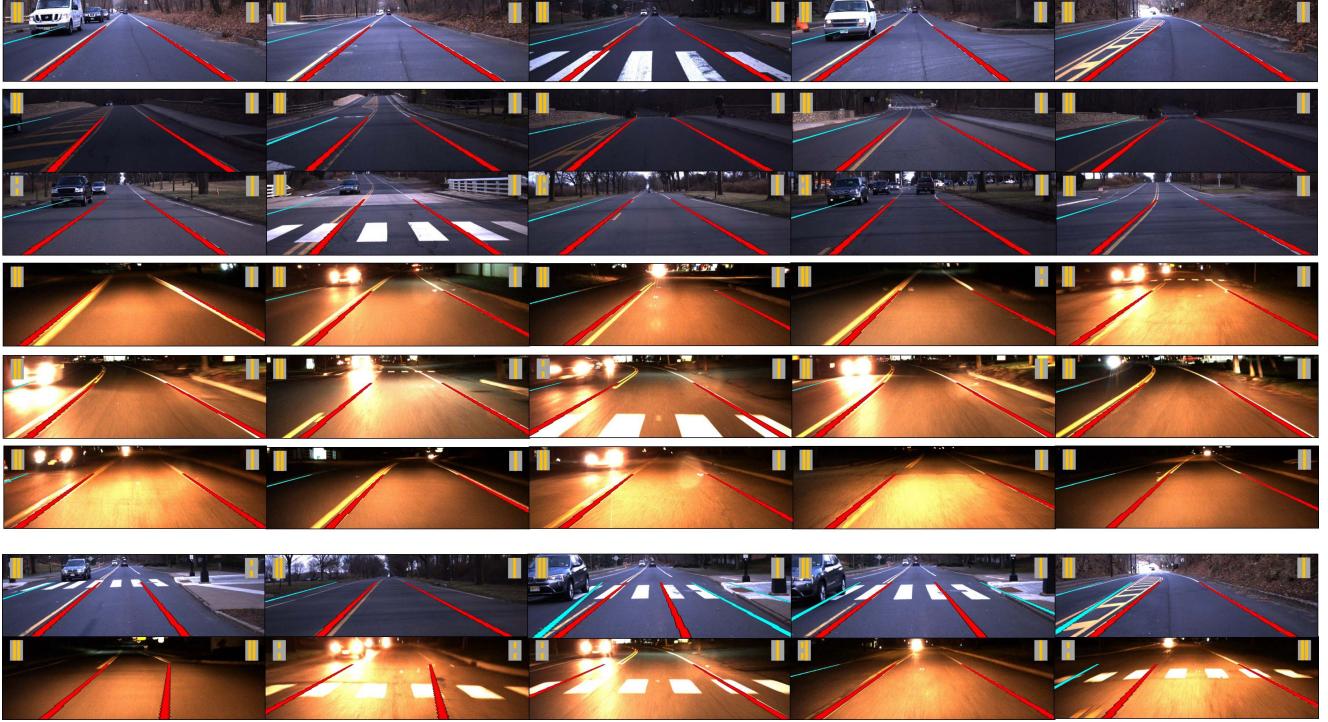


Fig. 18. Sample results of the experiments for our own dataset.

TABLE II
EVALUATION RESULTS FOR HOST LANE (WITH ITS ADJACENT LANES)
DETECTION AND HOST LANE CLASSIFICATION

Scenarios	H-TPR	A-TPR	A-FPR	H-CPR
KITTI	Road_0004	0.896	0.824	0.043
	Road_0015	1.00	0.723	0.00
	Road_0027	1.00	1.00	0.010
	Road_0028	0.993	1.00	0.077
	Road_0042	0.978	0.717	0.009
Ours	Daytime1	0.989	0.934	0.037
	Daytime2	1.00	0.957	0.008
	Night1	0.997	0.551	0.014
	Night2	0.993	0.615	0.003
	Night3	0.954	0.335	0.011

detected. **Target Adjacent** is the number of adjacent lanes that need to be detected. **Adjacent** is the number of all the adjacent lanes, including **Target Adjacent** and those that were not necessary to be detected, such as the completely blocked adjacent lanes, no-painted adjacent lanes, etc.

The presented system was tested on a machine with Intel Core i7 processor (3.1GHz, 4 cores), NVIDIA GeForce Graphics Card (GTX 650) and 4GB RAM. It could work effectively at a rate of 15 Hz to meet the requirement of the general driving speed (80km/h). The given run time includes all the modules in our system. In detail, the approximate calculation cost allocation is: disparity image calculation (75%), obstacles segmentation (3%), IPM and filtering (5%), Hough Transform and lane detection (9%), host lane classification and collision distance calculation (8%).

In general, for a driving ego-car, host lane and its adjacent lanes are always much more important than other farther lanes. What's more, it's easy to estimate the other lanes based on an electronic map or geographic information system (GIS)

once we detect host lane and its adjacent lanes. Therefore, we only focus on the evaluation for host lane and its adjacent lanes without evaluating for other farther lanes. As shown in TABLE II, we calculated host lane detection true positive rate **H-TPR**, adjacent lanes detection true positive rate **A-TPR**, adjacent lanes detection false positive rate **A-FPR** and host lane classification true positive rate **H-CPR** in our evaluation system to present the accuracy of lane detection and classification. To calculate these values for each dataset, we also counted other values. Specifically, DH is the number of the true detected side lines in all the target host lanes, SH is the number of the correct supplementary side lines in all the target host lanes, DA is the number of the true detected adjacent lanes among all the target adjacent lanes, FA is the number of the false detected adjacent lanes among all the adjacent lanes, CH is the number of true classified side lines among all the true detected host side lines. In this way, $\text{H-TPR} = (\text{DH} + \text{SH}) / (2 * \text{Target Host})$, $\text{A-TPR} = \text{DA} / (\text{Target Adjacent})$, $\text{A-FPR} = \text{FA} / \text{Adjacent}$, $\text{H-CPR} = \text{CH} / \text{DH}$. The evaluation results for these ten scenarios are shown in TABLE II and some sample detection and classification results are illustrated in Fig. 18 and Fig. 19. Specifically, Fig. 18 shows sample results of our own datasets. In detail, 1st row is Daytime1, 2nd and 3rd rows are Daytime2, 4th, 5th and 6th rows are Night1, Night 2 and Night3, respectively. Besides, some false sample results of our own dataset are presented in 7th and 8th rows. Fig. 19 presents the sample results of KITTI dataset. 1st, 2nd, 3rd, 4th, 5th rows in Fig. 19 are Road_0004, Road_0015, Road_0027, Road_0028, Road_0042 in KITTI dataset, respectively. What's more, 6th and 7th rows show some false sample results of KITTI dataset. It's obvious that the image quality of our own dataset is much poorer than KITTI dataset. Especially



Fig. 19. Sample results of the experiments for KITTI dataset.

in the night scenarios, the collected images were usually blurred and off-color because the exposure time increased. Besides, roads in our own dataset were always bumpy (uphill, downhill, valley, crest, ect.) compared with KITTI dataset, as shown in Fig. 18. The poor image quality and bumpy environments indeed increase great challenges for the common vision-based lane detection methods but exactly prove our system's robustness. However, it should be noted that we didn't use the obstacles segmentation and fusion module in our system for night experiments because we found the generated disparity image was full of noise sometimes with the image quality becoming worse at night.

In TABLE II, we can find **H-TPR** for different scenarios are usually very high (sometimes can reach 1.00). But it's only 0.896 for Road_0004 because there is a painted fake line on the road which is very close to the true line. **A-TPR** is also okay but relatively lower than **H-TPR** especially for the night scenarios. This is because adjacent lanes are usually influenced by the sunshine or shade in some scenarios. What's more, adjacent lanes are always submerged in darkness in the night scenarios because our car lamp is not particularly bright. In addition, **H-TPR** are always very low for different scenarios even though there is a lot of noise or interference in the datasets. For host lane classification, **H-CPR** is very high for these scenarios (always can reach approximately 0.99). However, it is only 0.863 for Road_0028. This is because the right-side line of host lane is usually influenced by the roadside weeds and their reflected light. In order to provide more information, we will public our dataset so that other stereo vision-based lane detection and classification researchers can compare their methods with ours according to the presented evaluation index.

Based on the stereo vision system, we developed a light-weight collision distance detection module which only aimed at providing a coarse collision warning function rather than accurate distance measurement as lidar does. As we introduce above, the created obstacle mask in our system is more

robust than a 2D lidar mask for the bumpy road although its detecting accuracy is much worse than the lidar. Therefore, the presented stereo vision-based collision detection method could be feasible for forward collision warning. To evaluate it, we calculated the front obstacles detection true positive rate (OTPR) and the front obstacles detection false positive rate (OFPR) in the host lane and its adjacent lanes. According to the testing results of our datasets (excluding the night scenarios), we found OTPR could achieve 100% and OFPR was about 6%. In other words, the designed detection sliding block could find front obstacles easily in the bird-view image but sometimes wrongly detected the nonexistent obstacles (because the disparity image or obstacles segmentation image was noisy sometimes).

VI. CONCLUSION AND FUTURE WORK

This paper presents a real-time and robust lane detection and classification method for forward collision warning system. Multiple techniques including stereo matching, obstacles segmentation, Hough Transform, deep learning and other vision-based algorithms are fused together in our system based on a self-adaptive traffic lanes model. Then, the detection and classification result is employed into a simple forward collision warning strategy, which may be very helpful for aided driving in some simple structured environments (like expressway). Experimental results in different scenarios based on two types of dataset proved that our method can realize high accuracy lane detection and classification, even in some challenging situations, such as the night, bumpy road, shadow, glare, etc.

For the future work, we intend to fuse this method with an electronic map or GIS to realize a better result. Meanwhile, through matching the observed road states with the priori electronic map information, the detection and classification result can be also used to locate the ego-car accurately at the lateral direction. Besides, the forward collision warning strategy also needs to be improved if we want to use it for daily driving.

REFERENCES

- [1] K. Bimbraw, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in *Proc. 12th Int. Conf. Informat. Control, Autom. Robot. (ICINCO)*, vol. 1, Jul. 2015, pp. 191–198.
- [2] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López, "Embedded real-time stereo estimation via semi-global matching on the GPU," *Procedia Comput. Sci.*, vol. 80, pp. 143–153, Jun. 2016.
- [3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.
- [4] J. Janai and F. Güney, A. Behl, and A. Geiger. (2017). "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art." [Online]. Available: <https://arxiv.org/abs/1704.05519>
- [5] S. Yenikaya, G. Yenikaya, and E. Düven, "Keeping the vehicle on the road: A survey on on-road lane detection systems," *ACM Comput. Surv.*, vol. 46, no. 1, 2013, Art. no. 2.
- [6] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, 2014.
- [7] A. M. Kumar and P. Simon, "Review of lane detection and tracking algorithms in advanced driver assistance system," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 4, pp. 65–78, Aug. 2015.
- [8] U. Ozgunalp, R. Fan, X. Ai, and N. Dahnoun, "Multiple lane detection algorithm based on novel dense vanishing point estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 621–632, Mar. 2016.
- [9] C. Li, Y. Nie, B. Dai, and T. Wu, "Multi-lane detection based on multiple vanishing points detection," *Proc. SPIE*, vol. 9443, p. 94430E, Mar. 2015.
- [10] H. Tan, Y. Zhou, Y. Zhu, D. Yao, and K. Li, "A novel curve lane detection based on improved river flow and RANSAC," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 133–138.
- [11] M. Beyeler, F. Mirus, and A. Verl, "Vision-based robust road lane detection in urban environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 4920–4925.
- [12] R. Gopalan, T. Hong, M. Shneier, and R. Chellappa, "A learning approach towards detection and tracking of lane markings," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1088–1098, Sep. 2012.
- [13] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust lane detection using two-stage feature extraction with curve fitting," *Pattern Recognit.*, vol. 59, pp. 225–233, Nov. 2016.
- [14] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004.
- [15] A. Borkar, M. Hayes, and M. T. Smith, "A novel lane detection system with efficient ground truth generation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 365–374, Mar. 2012.
- [16] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 289–295, Jan. 2016.
- [17] A. Das, S. S. Murthy, and U. Sudamalla, "Enhanced algorithm of automated ground truth generation and validation for lane detection system by M²BMT," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 996–1005, Apr. 2016.
- [18] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 7–12.
- [19] J. Wang, T. Mei, B. Kong, and H. Wei, "An approach of lane detection based on inverse perspective mapping," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 35–38.
- [20] S.-N. Kang, S. Lee, J. Hur, and S.-W. Seo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *Proc. IEEE Intell. Veh. Symp.*, Jan. 2014, pp. 221–226.
- [21] A. Linarth and E. Angelopoulou, "On feature templates for particle filter based lane detection," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1721–1726.
- [22] M. Fu, X. Wang, H. Ma, Y. Yang, and M. Wang, "Multi-lanes detection based on panoramic camera," in *Proc. 11th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2014, pp. 655–660.
- [23] A. Gurgelian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "DeepLanes: End-to-end lane position estimation using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun./Jul. 2016, pp. 38–45.
- [24] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2017.
- [25] B. He, R. Ai, Y. Yan, and X. Lang, "Accurate and robust lane detection based on dual-view convolutional neural network," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 1041–1046.
- [26] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *Proc. Int. Conf. Neural Inf. Process.*, 2014, pp. 454–461.
- [27] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.
- [28] M. Braga de Paula and C. R. Jung, "Automatic detection and classification of road lane markings using onboard vehicular cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3160–3169, Dec. 2015.
- [29] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 607–616, Sep. 2010.
- [30] B. Huval *et al.* (2015). "An empirical evaluation of deep learning on highway driving." [Online]. Available: <https://arxiv.org/abs/1504.01716>
- [31] M. Haloi and D. B. Jayagopi, "A robust lane detection and departure warning system," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun./Jul. 2015, pp. 126–131.
- [32] S. J. W. Tang, K. Y. Ng, B. H. Khoo, and J. Parkkinen, "Real-time lane detection and rear-end collision warning system on a mobile computing platform," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2015, pp. 563–568.
- [33] T. Kowsari, S. S. Beauchemin, and M. A. Bauer, "Map-based lane and obstacle-free area detection," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 3, Jan. 2014, pp. 523–530.
- [34] D. Levi, N. Garnett, E. Fetaya, and I. Herzlyia, "StixelNet: A deep convolutional network for obstacle detection and road segmentation," in *Proc. BMVC*, 2015, p. 109.
- [35] J. Yang, H. Wang, Z. Ding, Z. Lv, W. Wei, and H. Song, "Local stereo matching based on support weight with motion flow for dynamic scene," *IEEE Access*, vol. 4, pp. 4840–4847, 2016.
- [36] Y. Liu, J. Yang, Y. Lin, and H. Lv, "Research on medical applications of contrast sensitivity function to red-green gratings in 3D space," *Neurocomputing*, vol. 220, pp. 34–40, Jan. 2017.
- [37] M. Cordts *et al.*, "The stixel world: A medium-level representation of traffic scenes," *Image Vis. Comput.*, vol. 68, pp. 40–52, Dec. 2017.
- [38] J. Yang, Y. Lin, and Y. Liu, "Stereo chromatic contrast sensitivity model to blue-yellow gratings," *Opt. Exp.*, vol. 24, no. 5, pp. 4488–4496, 2016.
- [39] J. Yang, Z. Gao, R. Chu, Y. Liu, and Y. Lin, "New stereo shooting evaluation metric based on stereoscopic distortion and subjective perception," *Opt. Rev.*, vol. 22, no. 3, pp. 459–468, 2015.
- [40] Y. Gao, X. Ai, Y. Wang, J. Rarity, and N. Dahnoun, "U-V-disparity based obstacle detection with 3D camera and steerable filter," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 957–962.
- [41] J. Wang, Y. Wu, Z. Liang, and Y. Xi, "Lane detection based on random Hough transform on region of interest," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jun. 2010, pp. 1735–1740.
- [42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [43] W. Song, Y. Yang, M. Fu, F. Qiu, and M. Wang, "Real-time obstacles detection and status classification for collision warning in a vehicle active safety system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 758–773, Mar. 2018.
- [44] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [45] Y. Jia *et al.* (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: <https://arxiv.org/abs/1408.5093>



Wenjie Song received the B.S. degree in automation from the Beijing Institute of Technology, China, in 2013. He is currently pursuing the Ph.D. degree in intelligent robots with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, and also the School of Automation, Beijing Institute of Technology. Since 2016, he has also been with Princeton University as a Visiting Student.

His research interests include autonomous vehicle, computer vision, visual SLAM, path planning, and motion controlling. He has taken part in China Intelligent Vehicle Future Challenge several times as the Captain or Core Member.



Yi Yang (M'10) received the Ph.D. degree in control science and engineering with the School of Automation, Beijing Institute of Technology, Beijing, China.

He is currently an Associate Professor with the School of Automation, Beijing Institute of Technology. He is author/co-author of more than 40 conference and journal papers in the area of unmanned ground vehicle. His research interests are environment perception of unmanned ground vehicle, motion planning and control, and robot design and development.



Yujun Li received the B.S. degree in computer science from Shanghai Jiao Tong University, China, in 2014, where he is currently pursuing the Ph.D. degree in machine learning with the Department of Computer Science and Engineering. Since 2017, he has also been with Princeton University as a Visiting Student.

His research interests include deep reinforcement learning, stochastic optimization, and matrix computation.



Mengyin Fu received the B.S. degree from Liaoning University, China, the M.S. degree from the Beijing Institute of Technology, China, and the Ph.D. degree from the Chinese Academy of Sciences.

He was elected as the Yangtze River Scholar Distinguished Professor in 2009. He was a recipient of the Guanghua Engineering Science and Technology Award for Youth Award in 2010 and the National Science and Technology Progress Award for several times in recent years.

Prof. Fu is the President of the Nanjing University of Science and Technology. His research interest covers integrated navigation, intelligent navigation, image processing, learning, and recognition and their applications.



Meiling Wang received the B.S. degree in automation from the Beijing Institute of Technology, China, in 1992, and the M.S. and Ph.D. degrees from the Beijing Institute of Technology, China, in 1995 and 2007, respectively. She has been teaching at the Beijing Institute of Technology since 1995, and worked at the University of California San Diego as a Visiting Scholar in 2004.

She was elected as the Yangtze River Scholar Distinguished Professor in 2014. She is currently the Director of the Integrated Navigation and Intelligent Navigation Laboratory, Beijing Institute of Technology, China. Her research interests include advanced technology of sensing and detecting and vehicle intelligent navigation.

Navigation Laboratory, Beijing Institute of Technology, China. Her research interests include advanced technology of sensing and detecting and vehicle intelligent navigation.