ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/332438373

Robust Obstacle Detection and Recognition for Driver Assistance Systems

Article *in* IEEE Transactions on Intelligent Transportation Systems · April 2019

DOI: 10.1109/TITS.2019.2909275

CITATIONS
4

READS
358

1 author:

Jiaxu Leng
Chinese Academy of Sciences

18 PUBLICATIONS **56** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project object detection View project

Project object tracking View project

All content following this page was uploaded by Jiaxu Leng on 23 April 2019.

The user has requested enhancement of the downloaded file.

Robust Obstacle Detection and Recognition for Driver Assistance Systems

Jiaxu Leng*†, Ying Liu*†‡, Dawei Du*, Tianlin Zhang*†, Pei Quan*†

***Abstract*—This paper proposes a robust obstacle detection and recognition method for driver assistance systems. Different with the existing methods, our method aims to detect and recognize obstacles on the road rather than all the obstacles in the view. The proposed method involves two stages aiming an increased quality of the results. The first stage is to locate the positions of obstacles on the road. In order to accurately locate the on-road obstacles, we propose an obstacle detection method based on the U-V disparity map generated from a stereo vision system. The proposed U-V disparity algorithm makes use of the V-disparity map that provides a good representation of the geometric content of the road region to extract the road, and then detects the on-road obstacles using our proposed realistic U-disparity map that eliminates the foreshortening effects caused by the perspective projection of pinhole imaging. The proposed realistic U-disparity map greatly improves the detection accuracy of the distant obstacles compared with the conventional U-disparity map. Second, the detection results of our proposed U-V disparity algorithm are put into a context-aware Faster-RCNN that combines the interior and contextual features to improve the recognition accuracy of small and occluded obstacles. Specifically, we propose a context-aware module and apply it into the architecture of Faster-RCNN. Experimental results on two public datasets show that our proposed method achieves state-of-the-art performance under various driving conditions.**

***Index Terms*—Obstacle location, obstacle recognition, U-V disparity map, neural network, road extraction.**

I. Introduction

With the rapid development of the automobile industry, the driver assistance system (DAS) has become a hot research topic of the intelligent transportation system [24-27]. Obstacle detection and recognition, as one of the main tasks of DAS, can identify the danger of driving ahead of time, increase safety factor and reduce traffic accidents effectively. However, there still remain several great challenges in real scenarios, such as illumination change, partial occlusion and weather conditions.

In the DAS, we usually only focus on the obstacles on the road. It means that we should extract the road first, and then locate the candidate obstacle regions on the road. The quality of road extraction seriously affects the detection results. Most of the existing road detection methods have achieved good performance using monocular cameras. However, the computational complexities of these algorithms are too high to satisfy real-time requirements. In order to extract the road efficiently, Zhang et al. propose an obstacle detection method that makes use of the U-V disparity map [18]. The U-V disparity map proposed in [3] is an extension of Labayrades V-disparity map [4]. In the U-V disparity map, it is easy to extract the road and the obstacles from the original image. The road is represented as an oblique line in the V-disparity map, and obstacles are represented as horizontal lines in the U-disparity map, which significantly reduces the processing time of road and obstacle extraction. However, existing U-V disparity methods seriously rely on the performance of the Hough transform [5] used in the V-disparity map. In addition, these algorithms suffer from the distant obstacles, which may consider two obstacles as one. To better detect the obstacles on the road, we propose a robust obstacle detection method that can accurately locate all on-road obstacles using the disparity map obtained from a stereo vision system. Considering the drawbacks of the traditional Hough transform that focuses on taking the line containing the most points, we first develop a road extraction method that combines the Hough transform and a special weighted least square approximation to achieve accurate road extraction, rather than using the traditional Hough transform. Specifically, the Hough transform first provides an approximate region, and then the proposed weighted least square extracts the oblique line representing the road in the V-disparity map. This operation is effective to reduce or eliminate noise. Furthermore, we present a new map called realistic U-disparity map, which performs better than the U-disparity map on detecting the obstacles, especially the distant obstacles. In the realistic U-disparity map, the distant obstacles are projected into the 3D space, which recovers their spatial location information and eliminates the foreshortening effects caused by the perspective projection of pinhole imaging. Thus, the proposed method can accurately locate all on-road obstacles.

Obstacle recognition plays an important role in the DAS, which is helpful for understanding traffic scenes. Therefore, our second work is to recognize the obstacles. Reliably recognizing an obstacle requires a variety of information, including the objects fine-grained details and the context surrounding it. Context helps us understand the image! Apart from strong psychological evidence approving the context information is important for humans to recognize objects, many empirical studies in the field of computer vision have also suggested that recognition algorithms can be improved by proper modeling of context. It has been well recognized in the vision community for years that context information, or relations between objects, enhances object recognition [6][7][37]. Most of these methods were proposed before the prevalence of

* School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101400, China
†Data Mining and High Performance Computing Lab, Chinese Academy of Sciences, Beijing 101400, China
‡ Key Lab of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China
Corresponding author: Ying Liu (yingliu@ucas.ac.cn)

deep learning. Currently, there is no significant progress on exploiting context for recognition learning. Most methods still focused on recognizing objects separately. In this paper, we propose a context-aware module and incorporate it into Faster-RCNN [1]. Our proposed context-aware module is designed to learn the contextual information surrounding obstacles for accurate obstacle recognition, especially for small and occluded obstacle recognition.

The main contributions of our work are summarized as follows:

- We propose a robust obstacle detection method that consists of road extraction and obstacle location based on U-V disparity maps. Specifically, we develop a weighted least square method for road extraction in V-disparity maps, which overcomes the drawbacks of traditional Hough transform. Moreover, we present a special map called realistic U-disparity map, which is helpful to detect the distant obstacles on the road.
- We design a context-aware module for learning the contextual information, and it is effectively incorporated into the architecture of Faster-RCNN. Experimental results have shown that the proposed module improves the recognition accuracy, especially for small and occluded obstacles.

The rest of our paper is organized as follows: In Section II, we briefly review the related works on obstacle detection and recognition. In Section III, we introduce our obstacle detection and recognition methods. In Section IV, we conduct extensive experiments to verify the effectiveness of our approach, and compare it with other state-of-the-art approaches. Finally, we put forward the conclusion in Section V.

II. Related Works

Over the past decade, detecting and recognizing on-road obstacles have aroused the interest of researchers. A large number of algorithms [8][9][10][23] have been proposed to detect and recognize the on-road obstacles based on various sensors such as cameras, lidars and radars. Millimeter wave radars are used usually to detect the on-road obstacles, but it is insensitive to stationary and non-metal obstacles. Meanwhile, the expensive price makes it difficult to be used widely. In recent years, it is becoming more and more popular to use the lidars to detect the on-road obstacles. However, the lidar cannot handle small and black obstacles well and the high price limits its popularity. Visual obstacle detection and recognition methods set one or more cameras as the main sensor suite. The camera cheaper than other sensors offers a wide field of view. Therefore, in this paper, we propose a robust obstacle detection and recognition method by using stereo cameras.

Obstacle detection and recognition methods [30-34] can be divided into two types, learning-based and non-learning-based methods. Non-learning-based methods detect and recognize obstacles by hand-designed features, obstacle movement, and the relations between the sizes and the distances of obstacles [11-13]. However, it is quite challenging to design a robust hand-designed feature that can adapt to various driving conditions. In recent years, deep neural networks have achieved good performance in feature extraction and successfully been applied in many fields such as target tracking, speech recognition, and object detection. In the field of object detection, the representative algorithm is the region-based CNN (RCNN) [14] proposed by Girshick et al. RCNN is mainly composed of three parts. First, a large number of candidate regions are generated through the conventional selective search (SS) [2]. Then, feature extraction is performed through a convolutional neural network. Finally, the trained SVM is used to classify the candidate regions. RCNN achieved the best performance at the time. However, it is difficult to train a good model because it consists of multiple independent parts. In addition, the speed is a problem because feature extraction of each candidate region is operated separately. To solve the speed problem, Girshick et al. propose Fast-RCNN [16] based on RCNN, which reduces the computational complexity by directly projecting the candidate regions in the original image to the feature maps of the final convolution layer. This makes it possible to achieve feature extraction for all candidate regions with the forward convolution operation once. Furthermore, an end-to-end architecture for object detection, called Faster-RCNN [1], is proposed by Girshick et al. This method presents a region proposal network (RPN), which is used to accurately extract the candidate regions. Compared to the SS, the number of generated candidate regions is less while the quality is higher.

Although the above methods have achieved impressing performance, they still suffer from small and occluded objects. In this paper, we aim to develop an effective obstacle detection method, which accurately extracts the road and locates distant obstacles. In addition, we propose to use the contextual information to help us recognize obstacles, especially small and occluded obstacles.

III. Our approach

The proposed approach includes two main stages: obstacle detection and obstacle recognition, as shown in Fig. 1. First, we choose a stereo vision system to produce disparity maps for obstacle detection. The stereo vision system [17] is used to produce a sub-pixel disparity map under various driving conditions due to its high quality and possible real-time implementations. The resulting disparity maps are used as the input to the proposed obstacle detection method, which effectively utilizes the V-disparity map and U-disparity map to extract the road and on-road obstacles respectively. Second, the detected obstacles are then classified into different categories by the proposed context-aware Faster-RCNN model, which considers the contextual information to enhance the obstacle recognition.

A. Locating the on-road obstacles using U-V disparity map

Most existing obstacle detection methods are designed with the use of deep learning to generate the region proposals that are likely to be obstacles. However, these approaches are too time-consuming to satisfy real-time requirements. Recently, Faster-RCNN achieved an impressive performance for fast generating the region proposals by the region proposal network (RPN) [1]. The RPN aims to detect all objects in the image,

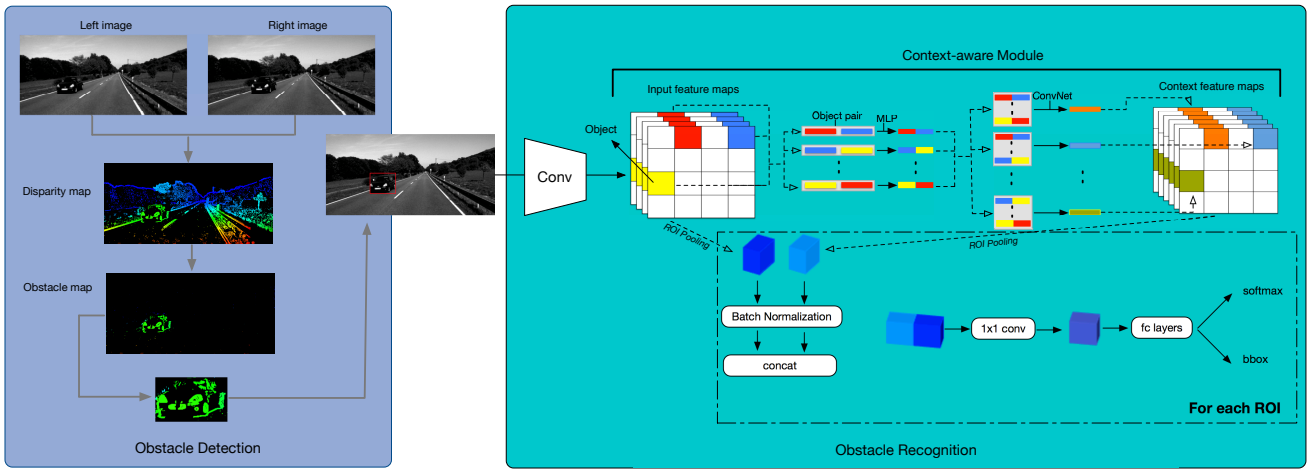


Fig. 1. The framework of our proposed obstacle detection and recognition method. The proposed method mainly consists of obstacle detection and recognition. First, we locate the candidate obstacle regions on the road using the disparity information. Then, the context-aware Faster-RCNN is developed to distinguish the categories of the candidate obstacles.

while the driver assistance system (DAS) only focuses on the on-road obstacles. In addition, the RPN fails to locate small and occluded obstacles.

In the DAS, we usually only focus on the obstacles on the road. Therefore, road extraction is the first step in our obstacle detection. The quality of road extraction seriously affects the detection results. Most of the existing road detection methods have achieved good performance using monocular cameras. However, the computational complexities of such algorithms are too high to satisfy the real-time requirements. In order to fast extract the road, the V-disparity map [19] is proposed by Labayrade et al., where V corresponds to the vertical coordinate of the original disparity map. Subsequently, Zhang et al. proposed an obstacle detection method by making use of the U-V disparity map [18]. Inspired by these methods, we present a robust obstacle detection method that utilizes the V-disparity map and U-disparity map to achieve road extraction and obstacle location, respectively.

The V-disparity map[19] and U-disparity map [20] are usually calculated by counting the number of pixels in rows or columns of a disparity map with the same disparity value. U-disparity (U_d) and V-disparity (V_d) are calculated as follows:

$$V_d(i,j)_{i\in H, j\in\Delta} = \sum_{n\in W} \varphi(i,j,n)$$
$$\varphi(i,j,n) = \begin{cases} 1 & \text{if } I_d(i,n) = j \\ 0 & \text{else} \end{cases} \tag{1}$$

$$U_d(i,j)_{i\in\Delta, j\in W} = \sum_{n\in H} \phi(i,j,n)$$
$$\phi(i,j,n) = \begin{cases} 1 & \text{if } I_d(n,j) = i \\ 0 & \text{else} \end{cases} \tag{2}$$

where $V_d(i,j)$ and $U_d(i,j)$ are the intensity value of pixel (i,j) in the V-disparity and U-disparity maps, respectively. H and W represent the original image width and height, respectively. Δ is the maximum disparity value. $\varphi(i,j,n)$ and $\phi(i,j,n)$ are two binary functions that are used to threshold the intensity values of the V-disparity and U-disparity maps. $I_d(i,n)$ and $I_d(n,j)$ are the intensity values of pixel (i,n) and (n,j) in the original disparity map, respectively.

The V-disparity map and U-disparity map are used to extract the road and obstacles respectively in our proposed obstacle detection method. Fig. 2 shows the block diagram of our method for obstacle detection, which consists of two steps: road extraction and obstacle location.

Road Extraction: When the road is plane, it is easy to extract the road using the Hough transform in the V-disparity, where the road is projected as an oblique line. However, the road is usually not plane in the actual scenes, instead, the middle part of the road is usually higher than both sides of the road. Therefore, the road cannot be simply represented by an oblique line, but a strip rectangular region. Generally, we consider the oblique line at the center of the strip rectangular region. The Hough transform focuses on taking the line containing the most points so that it may regard the diagonal line of the strip rectangular region as the road, which brings an incomplete road. To improve the accuracy of road extraction, we combine Hough transform and a special weighted least square. The Hough transform provides an approximate region, and then the weighted least square is used to extract the line representing the road, which is effective for reducing and eliminating noise. Specifically, we first detect the oblique line (the blue line in the V-disparity map of Fig. 2) based on the Hough transform in the V-disparity map. The detected oblique line is expressed as $v = kd + b$, where k and b are the slope and intercept of the line. Then, we collect the noteworthy pixels $(\bar{v}, \bar{d})$ surrounding the blue line that satisfy the following condition,

$$|\bar{v} - (k\bar{d} + b)| < \varepsilon \tag{3}$$

where ε is the maximum threshold of the distance between the point $(\bar{v}, \bar{d})$ and the oblique line detected by Hough transform. In this way, most noise is filtered, which is helpful for the subsequent road extraction. Furthermore, we provide an adaptive weight method that assigns different weights ω to the noteworthy pixels $(\bar{v}, \bar{d})$. Considering that there exist different

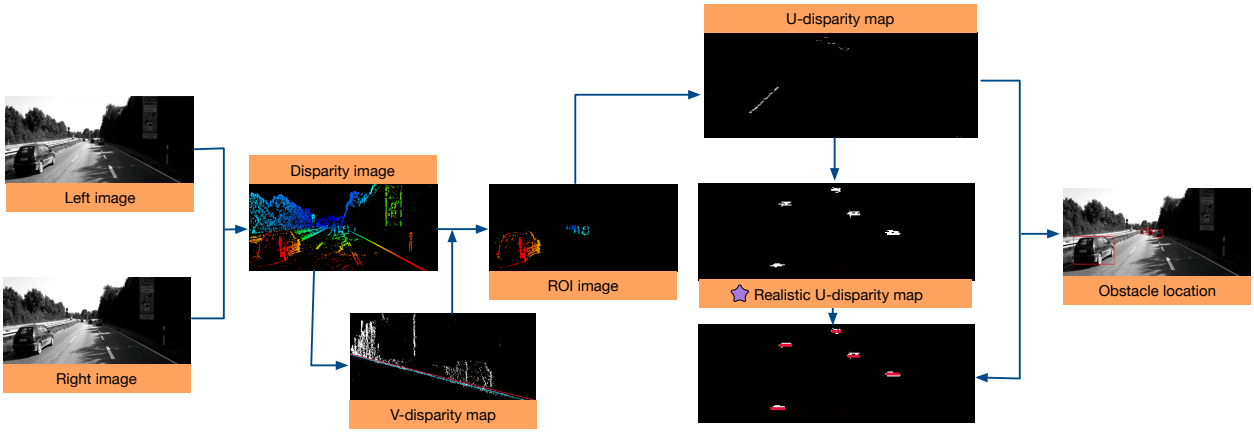


Fig. 2. The block diagram of the proposed obstacle detection method using the V-disparity map and U-disparity map. The road represented by the red line is detected in the V-disparity map using our proposed road detection method. The on-road obstacles are then detected by the proposed realistic U-disparity map, which projects the U-disparity map into the 3D scene.

stereo matching errors at different distances when generating the disparity map, the design of our weight assignment satisfies the principle: the larger the pixel value and the smaller the corresponding disparity value, the more likely it represents the road. ω is assigned to each noteworthy pixel as follows,

$$\omega(i,j) = \alpha\frac{i}{H} + (1-\alpha)\exp(v_d(i,j) - W) \tag{4}$$

where W and H are the width and height of the disparity map. α is a weighting factor considering the effects of the pixel's location and value. It can be seen that the pixel is less likely to be noise having a greater intensity value and smaller disparity value in Eq. (4). Finally, we use the weighted pixels to extract the oblique line representing the road (the red line in the V-disparity map in Fig. 3) by a weighted least squares method. To demonstrate the effectiveness of our road extraction method, we project the detected oblique line to the original disparity map to provid a comparison between our method and Hough transform. As shown in Fig. 3, our road extraction method achieves a better result.

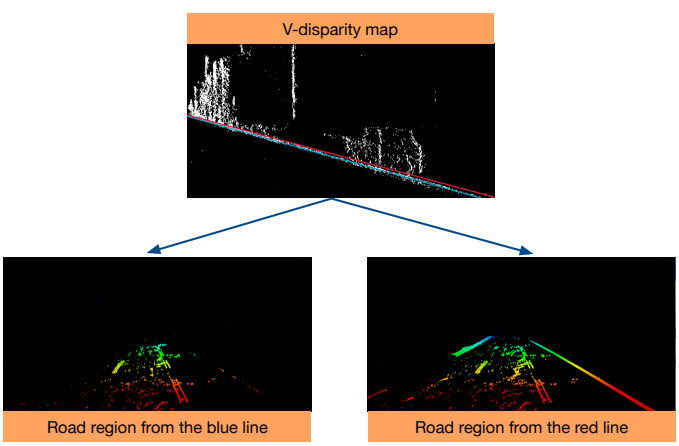


Fig. 3. Up: the blue line and red line are detected in the V-disparity map by Hough transform and our proposed method, respectively. Down: the right and left are two disparity maps only obtaining the road by projecting the red line and blue line to the disparity map, respectively.

Obstacle Location: The second step is to detect all obstacles on the road. Considering that the conventional U-disparity map cannot deal well with the distant obstacles, we present a new map, called realistic U-disparity map, by mapping the U-disparity map into the actual 3D scene.

In what follows, we will perform positioning in three coordinate systems shown in Fig. 4: R_w (absolute), R_{cr} (right camera), and R_{cl} (left camera). θ is the angle between the optical axis of the cameras and the horizontal. h is the height of the cameras above the ground and b is the distance between the cameras.

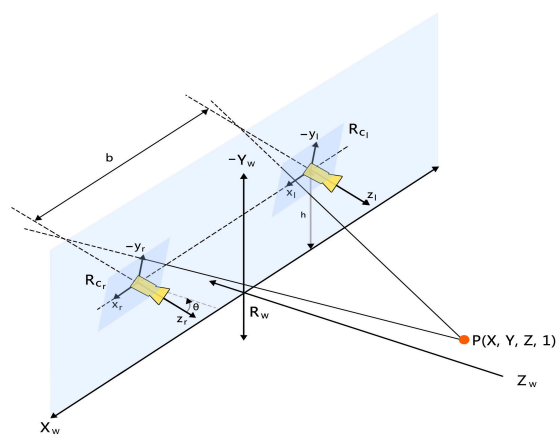


Fig. 4. The stereo sensor and the coordinate systems used

An arbitrary point $P(X,Y,Z)$ in world coordinate systems is mapped into an image plane through some stages. The mapping equations are expressed by:

$$\begin{cases} v_l = v_r = \frac{(f_v\cos\theta + v_0\sin\theta)(Y+h) + (v_0\cos\theta - f_v\sin\theta)Z}{Z\cos\theta + (Y+h)\sin\theta} \\ u_l = u_0 + f_u\frac{X+b/2}{Z\cos\theta+(Y+h)\sin\theta} \\ u_r = u_0 + f_u\frac{X-b/2}{Z\cos\theta+(Y+h)\sin\theta} \end{cases} \tag{5}$$

where (u_l, v_l) and (u_r, v_r) are the projection points of the point P. (u_0, v_0) is the center of image plane. f_u and f_v are the camera focal lengths in terms of pixel dimensions in u and v directions. In this research, θ is equal to 0 and the focal lengths

of two directions are same. Based on the above condition, our proposed realistic U-disparity map can be calculated as follows:

$$\begin{cases} Z = \frac{bf}{d} \\ X = \frac{(u-u_0)Z}{f} + \frac{1}{2}b \end{cases} \tag{6}$$

where X and Z are the horizontal and vertical coordinates of the realistic U-disparity map corresponding to u and d of the U-disparity map, and they denote the horizontal and vertical distances in the actual 3D scene, respectively. b and f are the camera parameters, denoting the distance of the two cameras and the focal length in terms of pixel dimensions respectively. Eq. (6) indicates that the U-disparity map is projected into the 3D scene, which clearly shows the actual position information of all the obstacles. As shown in Fig. 2, it is easier to detect the on-road obstacles compared with the original U-disparity map. In the realistic U-disparity map, the distant obstacles are projected into the 3D space, which recovers their spatial location information and eliminates the foreshortening effects caused by the perspective projection of pinhole imaging. All the obstacles are accurately detected in the proposed realistic U-disparity map as shown in Fig. 2, where the red lines represent the detected obstacles. Finally, we map these red lines into the original image, and then achieve the locations of all the on-road obstacles. The main workflow of our proposed obstacle detection method is shown in Algorithm 1.

Algorithm 1: Obstacle Detection

Input: the color image and corresponding disparity map
Output: bounding boxes of on-road obstacles in the color image

Generate the V-disparity map (V_d) and U-disparity map (U_d) using the input disparity map;
Extract the oblique line representing the road by combining the Hough transform and weighted least square in V_d;
Create the realistic U_d for accurately detecting distant obstacles by projecting the U_d to the actual 3D scene ;
Detect the horizontal lines (O_n) representing all obstacles in the realistic U_d;
for $i \leftarrow$ *1* ***to*** O_n **do**
  Compute the width W of the obstacle i corresponding to the width of the detected line in the realistic U_d;
  Determine the height H of the obstacle i by searching the pixels of the disparity map having the same disparity value with the detected obstacle i in the realistic U_d ;
  Compute the IoU of the obstacle i and the other obstacle j having the close disparity value;
  if $IoU > 0.5$ **then**
    Merge the obstacle i and j;
  Store the obstacle

B. Context-aware Faster-RCNN

Obstacle recognition is an important part of a DAS. Obstacle recognition results allow the DAS to better analyze the scene and give immediate feedback to the driver, which is very effective in preventing the occurrence of accidents. Lately, Faster-RCNN has achieved impressive performance in object detection. However, it hardly considers the contexts surrounding the objects. In addition, most of the amounts of region proposals generated by the RPN are not useful for a DAS that only focuses on the obstacles on the road.

Based on the above considerations, we present a novel detection and recognition method based on Faster-RCNN for accurately detecting and recognizing on-road obstacles, as shown in Fig. 1. We first use the proposed method described in Section III.A to generate the on-road region proposals instead of using the RPN. Then we propose a context-aware module to enhance the Faster-RCNN for better recognizing small and occluded obstacles appearing little features. In the following content, we will introduce the proposed context-aware module in detail.

The proposed context-aware module utilizes the contextual information to determine categories of the obstacles. It mainly consists of a multi-layer perceptron (MLP) with three layers and a convolution neural network (ConvNet) with two layers. The MLP is designed to learn the relations across all pairs of objects. After learning the relations between objects, a two-layer ConvNet is used to integrate each relation with another one. Finally, we obtain the context feature maps that are the same size with the input feature maps. Because the context feature maps contain rich context information, they are helpful to enhance the recognition accuracy, especially for small and occluded objects.

Our proposed context-aware module is shown in the details of Fig. 5. The input of our module is the feature maps obtained by convolving the image. Suppose we have obtained k feature maps of size $d \times d$, where k is the number of convolution kernels of the final convolutional layer. Each of the k-dimensional feature vectors in the $d \times d$ feature maps is labeled with an arbitrary coordinate revealing its relative spatial position. Considering the fact that the number of objects of different images is very different and it is difficult for us to know what image features form an object, each of the k-dimensional feature vectors in the $d \times d$ feature maps is considered as an object for our context-aware module as shown in Fig. 5. This means that an "object" can be the background, an actual object, a merger of objects or a merger of objects and backgrounds, which gives the model a great deal of flexibility.

To learn the relations between all the object pairs, a three-layer MLP is designed, which consists of 512 units per layer with ReLU non-linearities. Consider how an MLP would learn the relations. The MLP would first receive all objects from the object set simultaneously as its input. Then, the relations between each object pair will be learned by the MLP with its weight parameters. Specifically, we concatenate the feature vectors of object pairs to produce new feature vectors. By the MLP, we obtain $n \times (n-1)/2$ 512-dimensional feature vectors (where n is the number of objects), each of which

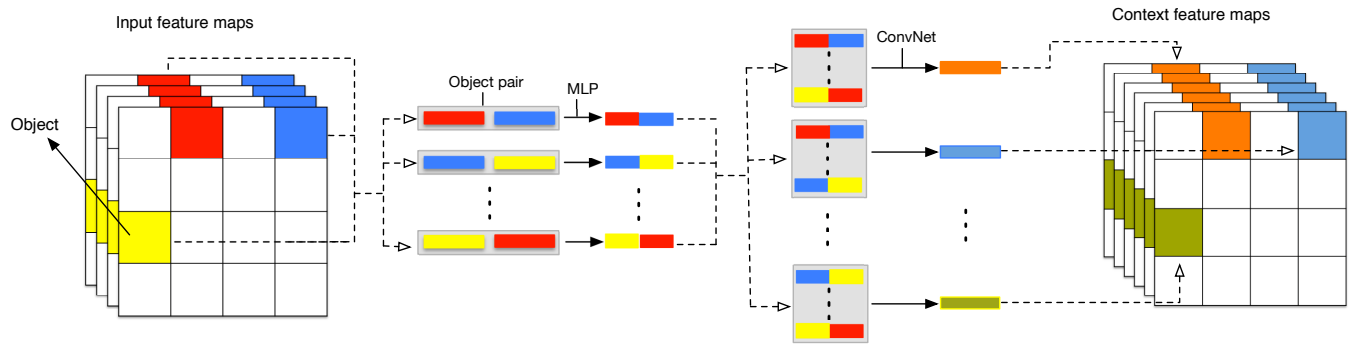


Fig. 5. The input is the feature maps obtained by convolving the image, and each cell is considered as an object. Three "objects" are colored here with red, blue, and yellow. The MLP considers relations across all pairs of objects, and then these relations are integrated by the ConvNet. Finally, the context feature maps with the same shape are obtained as the input. The context feature maps contain rich context information, which can be utilized to improve the performance of obstacle recognition

indicates the relation between two objects. The computing cost increases sharply as the number of objects grows. Since computing the relations between object pairs is independent, we can use GPU-based methods such as [35][36] to speed up the computation.

The MLP considers the object-level context, ignoring the image-level context. In addition, it is hard to apply the output of the MLP to existing networks. Our goal is to design a universal module, which can be applied to any existing network. In order to solve the two problems, we design a two-layer ConvNet to further learn the image-level context, while making the output have the same size with the input. For each object, we integrate all the relevant relations (512-dimensional feature vectors). Considering the influence of each object is different when determining the category of an object, we assign different weights learned through the ConvNet to different objects. Specifically, we obtain $(n-1)$ 512-dimensional feature vectors for each object by the MLP, and concatenate them to generate $(n-1)$ special feature maps with the size of 1×512. Then, we use two convolution layers with the kernel size of 1×3 to further learn the image-level context. The number of convolution kernels of two layers is 32 and 64, respectively. As a final note, we reduce the dimension with a 1×1 convolution, which produces an output with the same size as the input feature maps. Finally, every cell on output corresponds to every cell on input.

Fig. 1 shows how we integrate the context-aware module into Faster-RCNN. In a single pass, we use the proposed obstacle detection method described in Section III.A to generate region proposals, and then utilize the VGG16 [21] to extract features. According to the generated feature maps, our proposed context-aware module learns the context information, which produces context feature maps with the same size of the input feature maps. For each proposal, we obtain the local appearance features inside the proposal and the contextual features surrounding the proposal. We then use ROI pooling to produce a fixed-length feature descriptor for each proposal in the input feature maps and context feature maps, respectively. Each of descriptor pairs is then normalized with batch normalization, concatenated, and dimension-reduced (1×1 convolution) to produce a final descriptor. Each produced descriptor is processed by two fully-connected (fc) layers, and we finally obtain two outputs: a one-of-K class prediction, and an adjustment to the bounding box. The main workflow of our proposed obstacle recognition method is shown in Algorithm 2.

Algorithm 2: Obstacle Recognition

Input: the color image and ROIs
Output: positions and categories of on-road obstacles in the color image

1 Extract the features of the input color image using VGG16;
2 Map the positions of ROIs to the feature maps and context feature maps ;
3 **for** *each ROI* **do**
4 Extract the feature descriptor in the feature maps and context feature maps respectively;
5 Produce a fixed-length feature descriptor using the ROI pooling;
6 Descriptor pair is normalized, concatenated, and dimension-reduced to produce the final descriptor;
7 Obtain the category and adjusted position by two fully-connected layers ;
8 Store the category and position

IV. Experimental Results

A. Experimental Setup

Datasets and evaluation metric: To evaluate the effectiveness of our proposed method, we conducted extensive experiments on two datasets: the Karlsuhe Institute of Technology and Toyota Technology Institute (KITTI) dataset [22] and the CCD stereo camera dataset [27]. The two datasets provide abundant images under various driving conditions. To analyze our proposed method quantitatively, we use the detection rate (DR), false alarm rate (FA), and mean Average Precision (mAP) as our evaluation metrics.

Parameter Selection: In our proposed method, the parameter α in Eq. (4) adjusts the tradeoff between the pixel's location and the value for road extraction in the V-disparity

map. We conducted extensive experiments with various weight assignment conditions of the parameter α. Experimental results showed that our road extraction method is sensitive to α. We obtained different experimental results with different settings of α, which indicates that there exists a balance for the effects of the pixel's location and value. It is also worth noting that the parameter settings are associated with the quality of the disparity map. When the parameter α is set to 0.5, we achieve the best performance in these experiments. It means that the pixel's location and value make an equal impact on extracting the road.

The selection of the parameter ε in Eq. (3) also has a important influence on the quality of road extraction. ε is used to choose the pixels that represent the road in the V-disparity map. We conducted experiments to analyze the effect by varying the value of ε. Experimental results showed that our road extraction method is sensitive to the choice of ε. When the parameter ε is assigned a large value, the detected road area contained not only the real road but also some parts of on-road obstacles. When ε is assigned with a small value, we could only detect part of the road. In our experiments, we achieved the best performance for road extraction when ε is set to 4 pixels.

Implementation Details: The Python programming language is used to implement our proposed method under the framework of Tensorflow with GPUs. Our proposed method was implemented in Python. All of our experiments are carried out on two Intel Xeon processors with 64 Gb RAM. All experiments were conducted under the framework of Tensorflow with cuDNN v5.1.

To reduce the training time, we use the pre-trained weights of VGG16. The training process is divided into two phases. For the first phase, we trained the Faster-RCNN model with VGG16 by two steps. We first froze the pre-trained weights of VGG16, and trained the other weights of Faster-RCNN for 80k iterations, where the batch size and learning rate were 8 and 10^{-3} respectively. Then we released the frozen weights, and fine-tuned all the weights with a batch size of 32. The learning rate decayed from 10^{-3} to 10^{-5} by 10^{-1}. For each learning rate, we trained 120k, 40k, and 40k iterations separately.

For the second phase, we add our context-aware module to fine-tune the pre-trained Faster-RCNN model. Two steps are also conducted for training. We first froze the weights of Faster-RCNN in the first stage, and trained the extra weights of the context-aware module for 40k iterations, where the batch size and learning rate are 32 and 10^{-3} respectively. Then we released the frozen weights, and fine-tuned all the weights with a batch size of 32. The learning rate decayed from 10^{-4} to 10^{-6} by 10^{-1}. For each learning rate, we train 100k, 20k, and 20k iterations separately.

B. Results on Different Datasets

KITTI Dataset: KITTI provides a dataset under different driving conditions, including 7,481 training images and 7,518 test images, with a total of 80,256 labeled instances. In order to make the training and validation instances not overlap, we divided the KITTI dataset into two sub data sets: training data sets and validation data sets, and each sub data set contained 40,128 instances. We randomly selected 10,240 instances to train from the training data set, and 24,576 instances were tested from the validation data set respectively. This makes it more convincing to validate the effectiveness of our method.

Table I shows the experimental results of our method and Faster-RCNN under various training sizes. When using 2,048 samples to train our model, our proposed method achieves high accuracy with an mAP of 76.1%. Compared with the original Faster-RCNN, there is an improvement of 5.5% mAP, which benefits from the proposed obstacle detection method and context-aware module. The proposed obstacle detection method makes it possible to detect the distant obstacles on the road. In addition, our proposed context-aware module improves the recognition accuracy by making use of the contextual information. To further validate the effectiveness of our method, we conduct extensive experiments by increasing the training dataset size, and the experimental results are presented in Table I. It can be seen that the performance is improved when increasing the size of the training dataset. When the size of the training samples is increased to 10,240, our proposed method and Faster-RCNN provide 82.8% and 78.8% mAP, respectively. Evidently, our proposed method outperforms Faster-RCNN by 4.0% in mAP. The experimental results demonstrate the effectiveness of our proposed method.

To further validate the performance of our method, we compare it with the state-of-the-art detectors: Mono3D [30], NM-RDO [31], SubCNN [32], DPM-VOC+VP [33] and LSVM-MDPM-sv [34], which have shown to provide excellent performance in literature. We test our full detection and recognition pipeline on the KITTI dataset. Results are reported and compared with other advanced methods in Fig. 6. For each class, we present the experimental results in three regimes: easy, moderate and hard, which are defined according to the occlusion and truncation levels of obstacles. Fig. 6(a-c) shows the experimental results of different methods in easy, moderate and hard regime, respectively. It can be seen that our approach is superior in robustness compared to the other methods, especially in the hard regime. The success to perform well in the hard regimes is owed to our proposed realistic U-disparity map and context-aware module, which are useful to detect and recognize small and partially occluded obstacles. The Average Precision (AP) of each algorithm is presented in Table II. Among these algorithms, our algorithm achieves the best performance with 82.8% mAP, while the runtime is acceptable.

The CCD camera dataset: There are 50,100 labeled instances in the CCD stereo dataset. We again divide them into two sub data sets for training and validation. The training and validation data sets contain 25,050 instances respectively. We then use the two subsets to evaluate our method and compare with the baseline Faster-RCNN. Table III shows the results of our method and Faster-RCNN. It can be seen that our proposed method accurately detects and recognizes obstacles on the road, but Faster-RCNN raises more error detection. Compared with the Faster-RCNN, our method achieves a 4.2% improvement in mAP, which benefited from the proposed obstacle detection method and context-aware

TABLE I
THE AVERAGE PRECISION (AP) OF OUR PROPOSED METHOD AND FASTER-RCNN ON THE KITTI DATASET

Training size	Validation size	Faster-RCNN			Our method		
		Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
2,048	24,576	80.2%	74.1%	57.5%	83.4%	76.2%	68.7%
4,096	24,576	81.6%	75.8%	59.5%	84.2%	77.8%	72.6%
6,144	24,576	83.8%	76.9%	63.7%	86.4%	79.0%	77.3%
8,192	24,576	85.4%	78.1%	65.7%	87.1%	80.6%	78.6%
10,240	24,576	86.8%	79.4%	70.2%	88.4%	82.1%	77.9%

TABLE II
THE AVERAGE PRECISION (AP) OF DIFFERENT METHODS ON THE KITTI DATASET

Method	Dataset	Runtime per frame	Car			Pedestrian			Cyclist		
			Moderate	Easy	Hard	Moderate	Easy	Hard	Moderate	Easy	Hard
DPM-VOC+VP[33]	KITTI	8s	66.2%	80.4%	49.8%	44.8%	59.6%	40.3%	31.1%	43.6%	28.2%
LSVM-MDPM-sv[34]	KITTI	10s	57.4%	71.7%	46.5%	39.3%	51.7%	35.9%	29.2%	37.7%	27.5%
Mono3D[30]	KITTI	4.2s	87.8%	90.2%	78.1%	66.7%	77.3%	63.4%	63.8%	75.2%	58.9%
NMRDO[31]	KITTI	0.1s	58.4%	72.6%	47.5%	41.2%	52.2%	36.8%	30.8%	38.4%	28.3%
SubCNN [32]	KITTI	2s	88.8%	90.7%	79.2%	71.3%	83.1%	66.3%	70.7%	77.8%	62.7%
Ours	KITTI	0.2s	89.2%	91.2%	84.8%	83.4%	87.8%	75.1%	79.5%	84.0%	70.2%

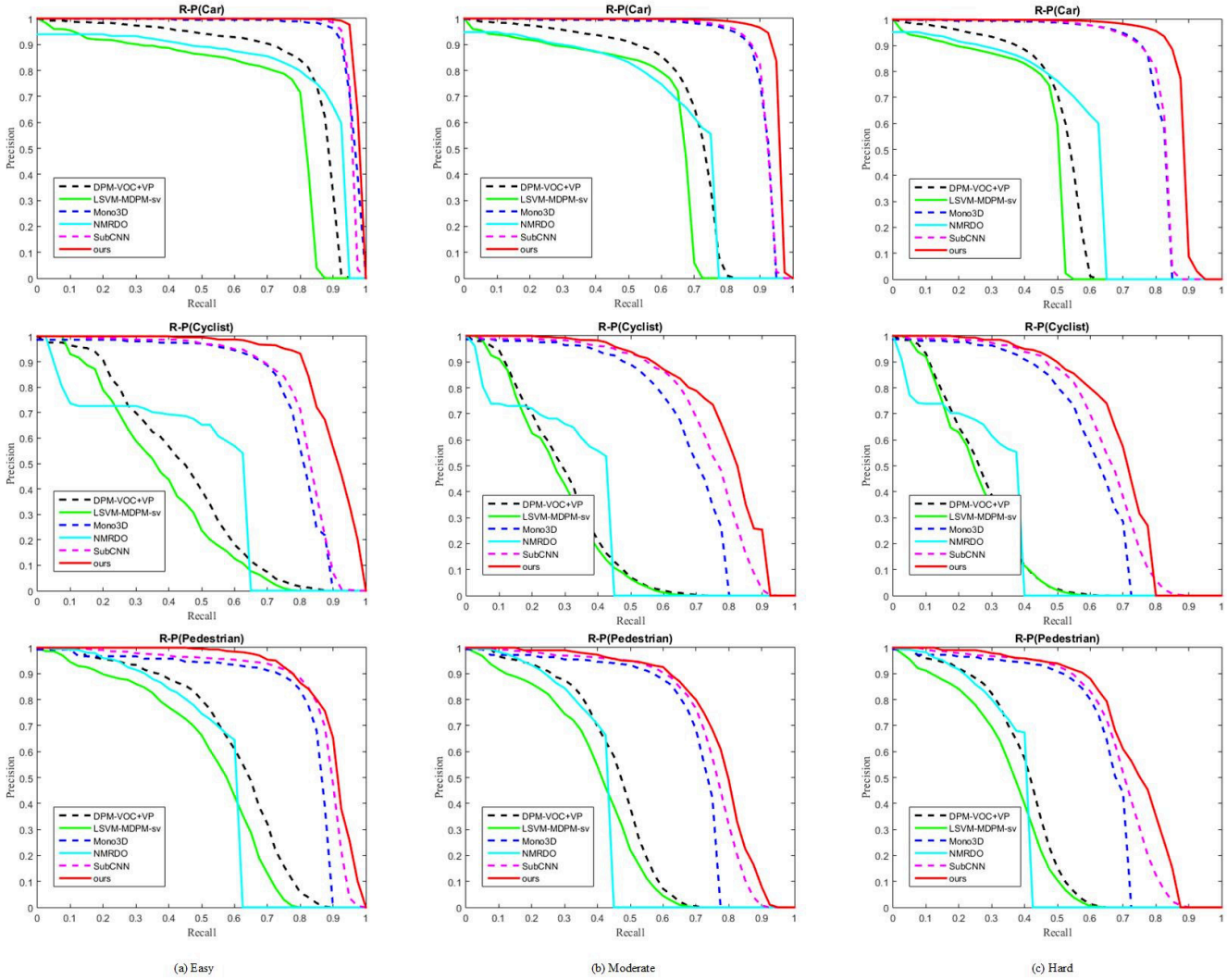


Fig. 6. Experimental results of our method and the state-of-the-art methods: Mono3D [30], NM-RDO [31], SubCNN [32], DPM-VOC+VP [33] and LSVM-MDPM-sv [34] on KITTI dataset.

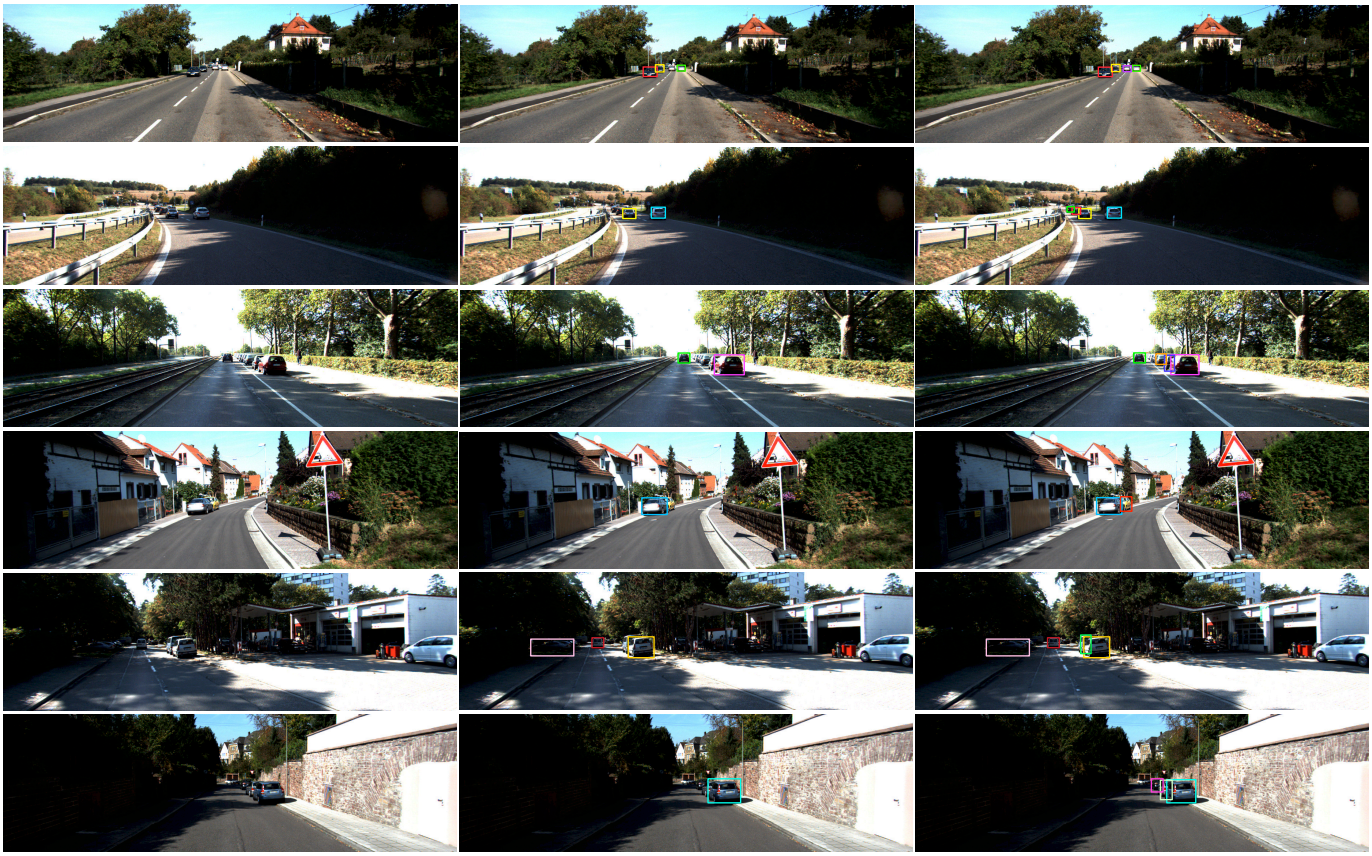


Fig. 7. Comparison between the conventional Faster-RCNN and our proposed method. The results of car detection and recognition with scores higher than 0.5 are shown. The input image is shown in the first column, and the results of the conventional Faster-RCNN and our method are shown in the second and third column respectively.

TABLE III
THE MEAN AVERAGE PRECISION (MAP) OF OUR PROPOSED METHOD AND FASTER-RCNN ON THE CCD STEREO DATASET

Training size	Validation size	Faster-RCNN	Our method
2,048	20,480	71.4%	76.5%
4,096	20,480	72.6%	78.9%
6,144	20,480	75.3%	81.2%
8,192	20,480	77.6%	82.6%
10,240	20,480	79.2%	83.4%

module. The proposed realistic U-disparity map improves the accuracy of detecting the distant obstacles by eliminating the foreshortening effects caused by the perspective projection of pinhole imaging. In addition, our proposed context-aware module improves the recognition accuracy of the on-road obstacles by making use of the contextual information. To further validate the effectiveness of our method, we conduct more experiments by increasing the training dataset size, and the experimental results were shown in Table III. It can be seen that the performance was improved by increasing the size of the training dataset. When increasing the size of training samples to 10,240, our proposed method and the Faster-RCNN provide 83.4% and 79.2% mAP, respectively. Compared to the Faster-RCNN, our proposed method achieves a better result with an improvement of 4.2% in mAP due to the proposed obstacle detection and context-aware module.

To further verify the effectiveness of the proposed approach, we conduct more experiments with different combinations of baselines, focal lengths and driving conditions [27], and compare with existing stereo-vision-based obstacle detection methods. Table IV shows the results of different obstacle detection methods. Our proposed method achieves the best performance with 98.17% average detection rate and 1.08 % average false alarm rate. The results of stereo vision and evolutionary algorithm(SEA) [27], feature-based system [24], stixel-based system [26], motion-based system [25] and multisource-based system [29] are listed in Table IV. Among the existing methods, the multisource-based system [29] achieves the best results with an average detection rate of 97.75% and an average false alarm rate of 1.25%. Compared with the multisource-based system, our method achieves a 0.42% improvement in detection rate, while reducing the false rate to 1.08%.

C. Ablation Studies

To verify the effectiveness of the proposed method, we explore the variants of our method and conduct experiments on the KITTI dataset.

First, we show the effect of our obstacle detection method based on U-V disparity maps. To this end, we use the original U-V disparity method and the improved U-V disparity method to replace the Region Proposal Network (RPN) of the Faster-RCNN. Experimental results show that our proposed U-V

TABLE IV
AVERAGE DETECTION RATES (DR) AND FALSE ALARM RATES (FA) OF DIFFERENT METHODS

Seq.	Illumination	images	SEA [27]		Faster-RCNN [1]		Feature-based system [24]		Stixel-based system [26]		Motion-based system [25]		Multisource-based system [29]		Ours	
			DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)	DR(%)	FA(%)
SQ01	Normal	110	100	0	98	0	92	8	95	1	93	3	100	0	100	0
SQ02	Normal	310	97	3	97	2	91	5	94	2	92	5	98	1	99	1
SQ03	Low,changed	76	90	3	93	3	78	7	85	6	82	6	95	1	96	1
SQ04	Normal	115	100	0	99	1	89	3	98	2	95	7	100	0	100	1
SQ05	Low	225	96	4	98	2	80	4	95	3	95	3	98	2	98	2
SQ06	High,changed	85	91	4	94	2	82	8	88	4	85	4	96	2	97	1
SQ07	Normal	197	98	2	98	2	88	4	98	2	96	2	98	1	98	0
SQ08	Normal	126	97	5	98	2	90	6	100	3	92	3	98	3	99	2
SQ09	High	597	98	2	98	2	79	7	93	2	92	5	98	1	97	1
SQ10	High	754	95	3	96	2	85	5	100	4	96	2	97	1	98	1
SQ11	High	800	94	4	95	2	87	11	98	5	96	2	97	2	97	2
SQ12	High	505	97	3	97	1	82	15	97	3	92	5	98	1	99	1
Average			96.08	2.75	96.75	1.75	85.25	6.92	95.58	3.08	91.75	4.25	97.75	1.25	**98.17**	**1.08**

disparity method achieves a better result with an improvement of 1.3% over the original U-V disparity method. The original U-V disparity method cannot detect the road exactly, which seriously affects the subsequent detection results. In addition, the original U-V disparity method fails to detect the distant obstacles using the U-disparity map. However, our proposed road extraction method improves the accuracy of detecting the road, and the proposed realistic U-disparity map makes it possible to detect the distant obstacles by projecting the U-disparity map to real 3D scene. The experimental results demonstrate the effectiveness of the proposed obstacle detection method.

To further demonstrate the effectiveness of the improved U-V disparity method for locating small and occluded obstacles on the road, we conduct experiments and compare the results with that of the original U-V disparity method [18]. As shown in Table VI, the improved U-V disparity method achieves a better result than the original U-V disparity method. There is an 11.88% improvement in recall rate, while the false alarm rate is lower than that of the original U-V disparity method. This benefits from the accurate road extraction and obstacle location using the proposed realistic U-disparity map. Fig. 7 shows that the improved U-V disparity method deal well with small and occluded obstacles on the road, which further illustrates the effectiveness of the improved U-V disparity method.

Next, we incorporate the context-aware module into the architecture of Faster-RCNN without RPN, and then combine the new architecture with two obstacle detection methods: the original U-V disparity method and the improved U-V disparity method, respectively, to validate the effectiveness of our proposed context-aware module. Two versions of our method are provided and we record their experimental results in Table V. The original Faster-RCNN is set as our baseline, and it provides an mAP of 78.8%. When adding the proposed context-aware module to the Faster R-CNN with the original U-V disparity method, it provides an mAP of 81.8%. Compared to the method without the context-aware module, there is a 3.7% improvement. The results benefit from the use of the contextual information, which enhances the recognition accuracy, especially for small and occluded obstacles. Furthermore, we add the proposed context-aware module into Faster-RCNN with the improved U-V disparity method and it achieves better performance with an mAP of 82.8%. This demonstrates the effectiveness of the proposed context-ware module.

We also try to analyze the effect of each part of our proposed context-aware module. The proposed context-aware consists of two sub-networks: a three-layer MLP and a two-layer ConvNet, which are utilized to learn the object-object and object-scene context, respectively. We run models with different settings and recorded their evaluations in Table VII. The MLP in the context-aware module is designed to learn the relations of object pairs. When adding the MLP to the Faster R-CNN framework, it provides an mAP of 81.4%. Compared to the baseline Faster-RCNN, there is a 2.6% improvement in mAP. The results benefit from the use of the relations between object pairs. Similarly, incorporating the ConveNet that is designed to merge the relations into the Faster-RCNN framework, we see that there is a performance enhancement of 1.6% compared to the baseline. This indicates that it is necessary to merge the relations for obstacle recognition. Finally, incorporating the MLP and ConvNet simultaneously to the Faster-RCNN, it provides the best performance with an mAP of 82.1%. There is a 3.3% improvement in mAP over the baseline. This further validates the rationality of the design of our context-aware module.

D. Qualitative Evaluation

We provide a qualitative comparison of our proposed context-aware Faster-RCNN with the conventional Faster-RCNN for detecting and recognizing on-road obstacles on the KITTI dataset, as shown in Fig. 7. It can be seen that our method achieves a better performance than the conventional Faster-RCNN, especially for small and occluded objects. Obviously, the conventional Faster-RCNN could not detect obstacles that suffer from partial occlusion. However, our context-aware Faster-RCNN is robust for occlusion as a result of the proposed context-aware module. This confirms that the recognition of an obstacle not only depends on the obstacle itself, but also on the surrounding contexts. In addition, our proposed method can handle small obstacles well, which is the weakness of conventional Faster-RCNN. The combination use of our realistic U-disparity map and context-aware module contributes to the success.

V. CONCLUSION

In this paper, we present an obstacle detection and recognition method, which utilizes the U-V disparity map and

TABLE V
ABLATION STUDY: THE EFFECTS OF THE IMPROVED U-V DISPARITY METHOD AND CONTEXT-AWARE MODULE

Method	Network	Dataset	Car	Pedestrian	Cyclist	mAP
Faster-RCNN	VGG16	KITTI	86.8%	79.4%	70.2%	78.8%
Faster-RCNN without RPN + U-V disparity method	VGG16	KITTI	86.7%	79.2%	69.8%	78.1%
Faster-RCNN without RPN + Improved U-V disparity method	VGG16	KITTI	87.0%	79.6%	70.4%	79.4%
Faster-RCNN without RPN + U-V disparity method + Context-aware module	VGG16	KITTI	87.6%	80.5%	71.3%	81.8%
Faster-RCNN without RPN + Improved U-V disparity method + Context-aware module	VGG16	KITTI	88.4%	82.1%	77.9%	82.8%

TABLE VI
ABLATION STUDY: THE EFFECTIVENESS OF THE IMPROVED U-V DISPARITY METHOD FOR LOCATING SMALL AND OCCLUSION OBSTACLES

Method	Dataset	IoU	Recall rate	False alarm rate
Improved U-V disparity method	KITTI	0.5	96.52%	80.42%
Original U-V disparity method	KITTI	0.5	84.64%	68.86%

TABLE VII
CONTRIBUTIONS OF EACH SUBNETWORK OF OUR CONTEXT-AWARE MODULE

Method	Network	Dataset	Car	Pedestrian	Cyclist	mAP
Faster-RCNN	VGG16	KITTI	86.8%	79.4%	70.2%	78.8%
Faster-RCNN + the MLP	VGG16	KITTI	88.4%	82.0%	73.8%	81.4%
Faster-RCNN + the ConvNet	VGG16	KITTI	87.4%	80.0%	71.8%	80.4%
Faster-RCNN + Context-aware module	VGG16	KITTI	88.7%	82.7%	74.9%	82.1%

contextual information to achieve state-of-the-art performance. The main contributions of the proposed method are as follows: 1) we improve the network structure of Faster-RCNN by adding an extra context-aware module that learns the contexts surrounding the objects. In this way, we can use contextual information to help us determine the categories of small and occluded obstacles; 2) we proposed an obstacle detection method that makes use of the U-V disparity map. The proposed obstacle detection can accurately detect on-road obstacles, especially distant obstacles; 3) we conduct extensive experiments on the KITTI and CCD camera datasets. The results show that our method achieves state-of-the-art performance.

The present work cannot handle the overlapped obstacles by using the non-maximum suppression (NMS). In future work, we will devote to solve this problem by improving the conventional NMS. In addition, we will consider how to transfer our proposed method into an embedded system so that it can be used in the DAS and produce commercial value.

ACKNOWLEDGMENT

This project was partially supported by Grants from Natural Science Foundation of China 71671178, 9154620 and the open project of the Key Lab of Big Data Mining and Knowledge Management. It was also supported by Guangdong Provincial Science and Technology Project 2016B010127004.

REFERENCES

[1] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(6): 1137-1149.

[2] Uijlings J R, De Sande K E, Gevers T, et al. Selective Search for Object Recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[3] Hu Z, Uchimura K. UV-disparity: an efficient algorithm for stereovision based scene analysis[C]//Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE, 2005: 48-54.

[4] Labayrade R, Aubert D, Tarel J P. Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation[C]//Intelligent Vehicle Symposium, 2002. IEEE. IEEE, 2002, 2: 646-651.

[5] Hough P V C. Machine analysis of bubble chamber pictures[C]//Conf. Proc. 1959, 590914: 554-558.

[6] Divvala S K, Hoiem D, Hays J H, et al. An empirical study of context in object detection[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 1271-1278.

[7] Galleguillos C, Rabinovich A, Belongie S. Object categorization using co-occurrence, location and appearance[C]//Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008: 1-8.

[8] Pantilie C D, Nedevschi S. Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow[C]//Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 2010: 439-444.

[9] Erbs F, Barth A, Franke U. Moving vehicle detection by optimal segmentation of the dynamic stixel world[C]//Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011: 951-956.

[10] Leng J, Liu Y. An enhanced SSD with feature fusion and visual reasoning for object detection[J]. Neural Computing and Applications, 1-10.

[11] Southall B, Bansal M, Eledath J. Real-time vehicle detection for highway driving[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 541-548.

[12] Chiu C C, Chung M, Chen W C. Real-Time Front Vehicle Detection Algorithm for an Asynchronous Binocular System[J]. J. Inf. Sci. Eng., 2010, 26(3): 735-752.

[13] Nishigaki M, Aloimonos Y. Moving obstacle detection using cameras for driver assistance system[C]//Intelligent Vehicles Symposium (IV), 2010 IEEE. IEEE, 2010: 805-812.

[14] Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 38(1): 142-158.

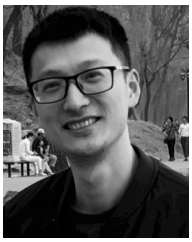
[15] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[C]//european conference on computer vision. Springer, Cham, 2014: 346-361.

[16] Girshick R B. Fast R-CNN[J]. international conference on computer vision, 2015: 1440-1448.

[17] Mayer N, Ilg E, Hausser P, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4040-4048.

[18] Zhang M, Liu P, Zhao X, et al. An obstacle detection algorithm based on UV disparity map analysis[C]//Information Theory and Information

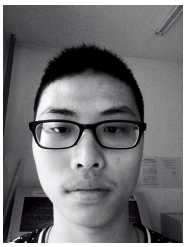
Security (ICITIS), 2010 IEEE International Conference on. IEEE, 2010: 763-766.
[19] Labayrade R, Aubert D, Tarel J P. Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation[C]//Intelligent Vehicle Symposium, 2002. IEEE. IEEE, 2002, 2: 646-651.
[20] Hu Z, Lamosa F, Uchimura K. A complete uv-disparity study for stereovision based 3d driving environment analysis[C]//3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on. IEEE, 2005: 204-211.
[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
[22] A. Geiger, P. Lenz, and R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn, Jun. 2012, pp. 3354-3361.
[23] Han J, Kim D, Lee M, et al. Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor[J]. IEEE Transactions on Vehicular Technology, 2012, 61(3): 971-985.
[24] Toulminet G, Bertozzi M, Mousset S, et al. Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis[J]. IEEE transactions on Image Processing, 2006, 15(8): 2364-2375.
[25] Pantilie C D, Nedevschi S. Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow[C]//Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 2010: 439-444.
[26] Erbs F, Barth A, Franke U. Moving vehicle detection by optimal segmentation of the dynamic stixel world[C]//Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011: 951-956.
[27] Nguyen V D, Nguyen T T, Nguyen D D, et al. A fast evolutionary algorithm for real-time vehicle detection[J]. IEEE Transactions on Vehicular Technology, 2013, 62(6): 2453-2468.
[28] Song W, Yang Y, Fu M, et al. Real-Time Obstacles Detection and Status Classification for Collision Warning in a Vehicle Active Safety System[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(3): 758-773.
[29] Nguyen V D, Van Nguyen H, Tran D T, et al. Learning framework for robust obstacle detection, recognition, and tracking[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(6): 1633-1646.
[30] Chen X, Kundu K, Zhang Z, et al. Monocular 3d object detection for autonomous driving[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2147-2156.
[31] Cheng M M, Zhang Z, Lin W Y, et al. BING: Binarized normed gradients for objectness estimation at 300fps[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 3286-3293.
[32] Xiang Y, Choi W, Lin Y, et al. Subcategory-aware convolutional neural networks for object proposals and detection[C]//Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. IEEE, 2017: 924-933.
[33] Pepik B, Stark M, Gehler P, et al. Multi-view and 3d deformable part models[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(11): 2232-2245.
[34] Geiger A, Wojek C, Urtasun R. Joint 3d estimation of objects and scene layout[C]//Advances in Neural Information Processing Systems. 2011: 1467-1475.
[35] Fialka O, Cadik M. FFT and convolution performance in image filtering on GPU[C]//Information Visualization, 2006. IV 2006. Tenth International Conference on. IEEE, 2006: 609-614.
[36] Kazmi M, Aziz A, Akhtar P, et al. A low cost structurally optimized design for diverse filter types[J]. PloS one, 2016, 11(11): e0166056.
[37] Hu H, Gu J, Zhang Z, et al. Relation networks for object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 3588-3597.



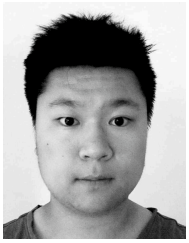
Jiaxu Leng received the B.S. degree in electronical information science and technology from Tianshui Normal University, Gansu, China, in 2012 and the M.S. degree in electrical engineering from University of Electronic Science and Technology, Sichuan, China, in 2015. After obtaining his master degree, he worked at Hisense for two years as algorithm engineer. He is currently pursuing the Ph.D. degree in computer science in University of Chinese Academy of Sciences, Beijing, China. His current research interests include image processing, object detection, object recognition, object tracking and stereo vision.

Ying Liu received the B.S. degree in computer science from Peking University, Beijing, China, in 1999. She received the M.S. and Ph.D. in computer engineering from Northwestern University, Evanston, IL, USA, in 2001 and 2005. She is professor in the University of Chinese Academy of Sciences and adjunct professor at Key Lab of Big Data Mining and Knowledge Management. Her current research interests include data mining, parallel computing and big data.

Dawei Du received the B.Eng. degree in automation and the M.S. degree in detection technology and automatic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree in University of Chinese Academy of Sciences, Beijing, China. His current research interests include object tracking, image processing and deep learning.



Tianlin Zhang received the B.S degrees in School of software of Nankai university, Tianjin, China, in 2017. He is currently pursuing the Master degree in University of Chinese Academy of Sciences, China. His current research interests include data mining and natural language processing.



Pei Quan received B.S. degrees in software engineering from Chongqing university, Chongqing, China, in 2017. He is currently pursuing the Ph.D degree in University of Chinese Academy of Sciences, China. His current research interests include data mining and image caption.

View publication stats