2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

# Exploring the Potential of Using Semantic Context and Common Sense in On-Road Vehicle Detection

Zhixiong Nan, Menghan Pan, Xiao Wang, Ping Wei*, Linhai Xu, Hongbin Sun, Jingmin Xin, and Nanning Zheng

*Abstract*— **Vehicle detection is an important research topic for autonomous driving community. Since the great success of deep learning on object detection, almost all vehicle detection methods go along with this line. However, deep learning methods heavily rely on the training data, and the whole mechanism is like a "black box". Therefore, in this paper, we explore a vehicle detection method using traffic semantic context and human common sense instead of relying on the training data. To verify our idea, we compare our method with two classic machine learning methods as well as three state-of-the-art deep learning methods on a dataset collected in real traffics. The results show that our method outperforms others on this dataset. The deep learning methods may exceed ours after enlarging the training data or testing on more complicated datasets. However, the main contribution of this paper is providing inspiration for learning methods, and we believe their performance can be greatly improved after considering the idea of this paper.**

## I. INTRODUCTION

Vehicle detection serves as a key technology of both autonomous vehicles and advanced driver assistance systems, and is significant for collision warning, collision avoidance, and accident reduction. Generally, vehicle detection consists of two stages, namely, candidate generation and candidate verification. In its early days, vehicle candidates are generated using some low-level or heuristic features, and are usually verified by the template-matching mechanism. At that time, the focus lies in finding good features. The features like the symmetry [1], [10], shadow [3], [13], and edge [21], [23] are widely used. Some symbols, such as tail lights, brake lights and tyres, are also employed for generating candidates [9]. However, these low-level features are sensitive to dynamic traffic scenes that can not meet the requirements of on-road vehicle detection. For example, some vehicles (*e.g.* overtaking vehicles and partly visible vehicles) usually present weak symmetry, thus are hard to be detected using symmetry feature. Shadow is easily influenced by illumination conditions. With the great success of haar-like fearture [25] for face detection in 2004 and HOG feature [4] for pedestrian detection in 2005, some high-level appearance features have been gradually applied for vehicle detection [7], [17]. However, traffic scenes are so complicated that these classic machine learning methods hardly behave well in dynamic traffic situations. For example, Haar-like is sensitive to illumination variations and dynamic environmental backgrounds. HOG [4] usually produces a large number of false positives. DPM [7] slightly contributes to performance improvement for small-scale vehicles that generally exhibit weak partial structures. More details about the development of vehicle detction techniques can be found in review articles such as [14], [22], [24].

Since 2012, deep learning has been actively pursued [18], [20], [30]. Deep learning based methods are more robust than traditional machine learning methods, and have achieved excellent detection accuracy on diverse vision tasks including vehicle detection. However, it still presents some deficiencies. Firstly, the performance of these methods is almost linear with the quality and quantity of training datasets, leading to the unexpected performance descent in unknown scenes if no similar samples are included in the training data. Secondly, deep learning paradigm is like a "black box" that we have not opened. Thirdly, machine learning based vehicle detection methods are usually subject to a 2D image plane where all information of a vehicle is assumed to be contained within a small 2D box, and few cues in 3D space are utilized. Factually, traffic scenes often feature high variations in weather, vehicle types, vehicle scales, illumination conditions, and environmental backgrounds. As a result, the limited number of training data can hardly accommodate diverse traffic conditions, and the idea of improving the performance by enlarging training datasets or deepening the neutral networks has nearly reached its limitation. In addition, as we know, there exists information loss during the procedure of projecting a 3D scene into a 2D image, thus exploring cues in 3D space is necessary.

To further boost the detection performance, it is time for us to imitate the human vision system. A driver effortlessly detects vehicles on roads since the driver understands the overall semantic scene and never regards a vehicle as an isolated pixel patch without information exchange with the surrounding pixels. In addition, the driver has some common sense about vehicular relative location, physical size, and its semantic interdependence with other objects. Moreover, a driver perceives the real 3D traffic which consists of more informative cues than a 2D image does. Factually, some pioneers have realized and explored the potential of overall semantic scene understanding in diverse vision tasks [8], [16], [27], [28] and some cognition-based studies [5], [29]

This work was supported partly by National Key R&D Program of China (No. 2017YFC0803907), National Natural Science Foundation of China (No. 61790563), and Joint Foundation of Ministry of Education of China (No. 6141A02033303)

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049 P.R. China

*Corresponding author: Ping Wei, pingwei@xjtu.edu.cn

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

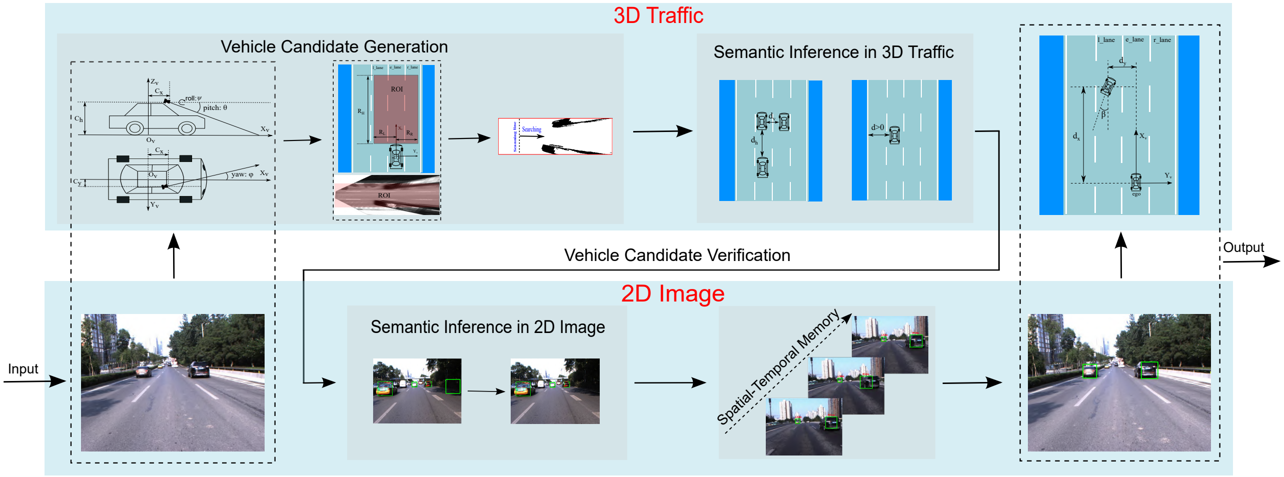

3D Traffic
Vehicle Candidate Generation
ROI
Searching
Semantic Inference in 3D Traffic
Vehicle Candidate Verification
2D Image
Input
Semantic Inference in 2D Image
Spatial-Temporal Memory
Output

Fig. 1: Overview of the proposed method.

have also proven that semantic cues are important for humans to detect objects.

Inspired by these observations, we explore to put vehicle detection into an overall scene understanding framework by involving 2D/3D traffic semantic context and human common sense. We believe the performance of existing on-road vehicle detection methods can be greatly improved after considering these cues. To validate our idea, we did not graft these cues onto the existing methods that are usually subject to a 2D image plane. Instead, we involve these cues into a 2D/3D vehicle detection framework where vehicle candidates are first generated in the context of 3D traffic, and then the candidates are verified by 2D/3D semantic and structural inference, which is realized by integrating the multiple cues of the geometric structure, priori knowledge, vehicle-object interdependence, vehicle interactions and appearance together with the spatial-temporal memory update. We annotated and released a dataset which is captured in real traffics. The proposed method is compared with five methods, namely, the HOG-SVM [4], DPM-LSVM [7], Faster R-CNN [20], SSD [12], and YOLO V2 [19] on this dataset. Experiment results show that our method behaves better than these learning based methods, which demonstrates that traffic semantic context and human common sense are important for vehicle detection.

## II. Approach

### A. Overview

The framework mainly consists of vehicle candidate generation and candidate verification. As Fig. 1 shows, a fast line-scanning searching strategy is firstly performed in 3D traffic to generate vehicle candidates. Subsequently, the semantic inference between 2D image and 3D traffic is conducted to verify the candidates, the semantic inference in 3D traffic is based on the analysis of the vehicle-object interdependence and vehicle interactions, while the semantic inference in 2D image utilizes the vehicle appearance. Afterwards, spatial-temporal memory is used to compensate for occasional missing detections. Finally, vehicles that contain rich information (*e.g.* the distance, velocity, accelerated velocity, inclination angle and lane label) are outputted. In this section, we first introduce the 3D traffic scene model which is the basis of the following two subsections: vehicle candidate generation and vehicle candidate verification.

### B. 3D Traffic Scene Model

To generate vehicle candidates in the context of 3D traffic and perform semantic inference in 3D traffic, the foremost issue is to build a 3D traffic scene model. We extend the viewpoint model and object model introduced in [8], obtaining a more informative 3D traffic scene model $M$ which is defined as

$$M = \{R, V, C\}. \tag{1}$$

As illustrated in Fig. 2 , the model $M$ contains three submodels, namely, the road surface model $R$, vehicle model $V$ and camera model $C$.

The road surface model $R$ is defined on the ground plane $Z_v = 0$ as a rectangular area

$$R = \{R_L, R_R, R_H\}, \tag{2}$$

where $R_L$ and $R_R$ describe the lateral scope of $R$, while $R_H$ describes the longitudinal scope.

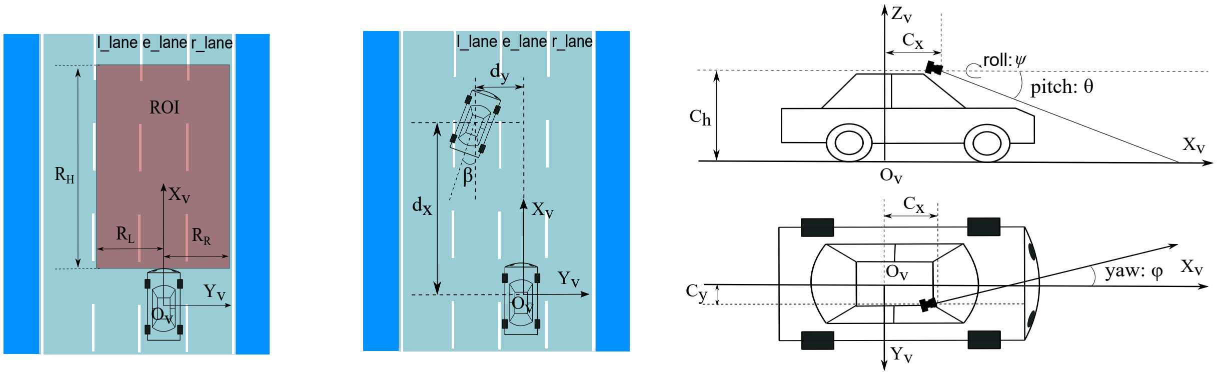
The vehicle model $V$ is defined as

$$V = \{d_x, d_y, u_x, u_y, a_x, a_y, \beta, l_l\}, \tag{3}$$

where $(d_x, d_y)$ denotes the relative location of a vehicle with respect to the ego-vehicle, $u_x$ and $u_y$ denote the velocities of a vehicle in longitudinal and lateral directions respectively, $a_x$ and $a_y$ denote the accelerated velocities in the corresponding directions, $\beta$ denotes the vehicle body inclination angle, and $l_l \subset \{left, ego, right\}$ denotes the lane label of a vehicle.

The camera model $C$, referring to the work [11], is defined as

$$C = \{C_x, C_y, C_h, \theta, \varphi, \psi\}, \tag{4}$$



(a) (b) (c)

Fig. 2: Visualization of 3D traffic scene model. (a) Road surface model. (b) Vehicle model. (c) Camera model.

where $(C_x, C_y, C_h)$ denotes the relative location of the camera with respect to the ego-vehicle origin $O_v$, $\theta$ denotes the pitch angle of the camera, $\varphi$ the yaw angle, and $\psi$ the roll angle.

### C. Vehicle Candidate Generation

The goal of this section is generating vehicle candidates in the context of 3D traffic. The major insights are (1) using the interdependence between vehicles and road surface to guarantee that the detected bounding boxes are with reasonable locations, (2) using human common sense about the vehicle size in the actual 3D world to guarantee that the detected bounding boxes are with reasonable scales, and (3) developing a fast vehicle candidate generation mechanism by narrowing down the candidate searching region and avoiding the repeated local searching.

Fig. 3 shows the vehicle candidate generation procedure. Firstly, the searching region image, as Fig. 3b shows, is jointly computed by the original image, camera model $C$ and road surface model $R$. Subsequently, an adaptive multi-threshold method is operated on the searching region image to obtain the binary searching region image as shown in Fig. 3c, on which a fast line-scanning searching strategy is finally applied to output vehicle candidates.

**Searching Region:** Different from traditional candidate generation methods that usually proceed in 2D image, the proposed method is performed within the candidate searching region in 3D traffic. The candidate searching region is decided by the road surface model $R = \{R_L, R_R, R_H\}$, where we define $R_H = 50m$ that characterizes the longitudinal scope ranging from 0 to $50m$, while $R_L$ and $R_R$ characterize the lateral scope spanning across the ego lane as well as the adjacent lanes, and $R_L$ and $R_R$ are obtained by expanding the ego lane markings that are automatically detected using the lane marking detection method introduced in [15]. Subsequently, the searching region image as shown in Fig. 3b is obtained by associating the candidate searching region $R$ with the original image. The relationship between the original image and $R$ is obtained using the camera model $C$ based on the flat road assumption. The camera model $C$ is resolved using the method introduced in [11].

The searching region image confirmation mechanism prevents the algorithm from searching invalid areas by using the interdependence between vehicles and the road surface, which improves algorithm efficiency and reduces the false candidates that appear in dead-zones like the sky, green belts, and buildings.

**Adaptive Multi-threshold Binarization:** To apply the searching strategy, the searching region image is binarized firstly. The original idea is to binarize the searching region image using a single threshold. However, we observed in the experiments that the single-threshold-based binary method exhibits poor performance if the intensity of ego lane differs from that of adjacent lanes. Particularly, when the ego-vehicle is driving on the leftmost or rightmost lane, roadside objects often result in the inhomogeneous intensity, leading to the inapplicability of the single-threshold-based binary method. Therefore, an adaptive multi-threshold binary method is proposed. As shown in Fig. 3b, the searching region image is evenly cut into three regions, namely, left lane region, ego lane region and right lane region. Three thresholds, $T_l$, $T_e$, and $T_r$, are separately computed for the left, ego, and right lane region using an iterative thresholding method introduced in [2]. With the thresholds, the searching region image (as Fig. 3b shows) is binarized, obtaining the binary searching region image (as Fig. 3c shows).

**Fast Line-scanning Searching Strategy:** The scanning line searches on the binary searching region image column by column for vehicle candidates. As illustrated in Fig. 3c, in each column, a group of continuous "black" pixels is regarded as a candidate if the number of the continuous "black" pixels belongs to an interval $I$. The interval $I$ is defined based on the priori knowledge of the vehicle size in the actual 3D world. The vehicle size is an important cue for vehicle detection. However, it is complicated to use this cue in 2D image because the vehicle size in image plane varies considerably with its distance. On the contrary, the vehicle size in the actual 3D world is relatively stable, using this common sense in 3D space guarantees that all detected bounding boxes in 2D image are with reasonable scales.

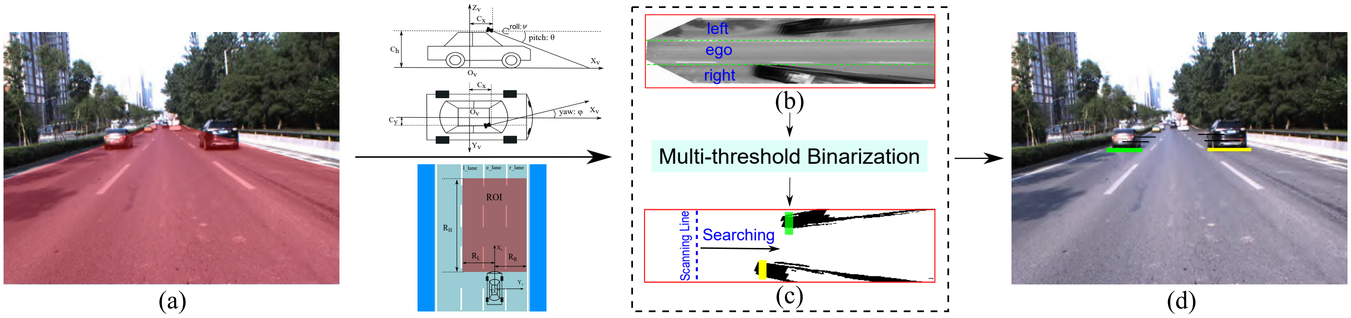


Fig. 3: Vehicle candidate generation procedure. (a) Original image. (b) Searching region image. (c) Binary searching region image. (d) Vehicle candidates.

### *D. Vehicle Candidate Verification*

To verify vehicle candidates and output coherent results, the semantic inference and spatial-temporal memory are utilized.

**Semantic Inference in 3D Traffic:** The vehicle interactions, which are measured by the safety distance between two vehicles, are used to reduce false candidates. The searching strategy discussed above generally outputs numerous candidates for one target because a vehicle often occupies multiple columns in the binary searching region image. By considering the fact that two vehicles maintain a safety distance $d_h$ in longitudinal direction, these false candidates are removed. Once a candidate is detected, its neighbouring longitudinal region within $d_h$ in the corresponding lane is regarded unavailable. Additionally, for two vehicles with similar longitudinal distances, the safety distance $d_v$ in lateral direction is considered. In this case, if the lateral distance of two candidates is less than $d_v$, the candidate in the ego lane is retained and the other candidate is removed.

The interdependence between vehicles and road boundaries is coarsely estimated to reduce the false candidates that locate beyond road boundaries. We observed that false candidates likely appear on the roadside if the ego-vehicle is driving on the leftmost or rightmost lane. To coarsely estimate whether a vehicle candidate is beyond road boundaries, a single threshold $T_e$ is used to binarize the whole searching region image. Subsequently, we sample several rows of the binarized searching region image, and count the "black" pixels in each sampled row. If the majority of one sampled row is occupied by the "black" pixels, the corresponding lane is regarded unavailable and the candidates in this lane are regarded locating beyond road boundaries.

**Semantic Inference in 2D Image:** After the semantic inference in 3D traffic, vehicle candidates are then projected into the image plane under the assumption that detected boxes are square, obtaining some candidate bounding boxes. The visual cues in 2D image are further used to refine the candidates. Based on the fact that a vehicle generally contains at least one salient horizontal structure, a candidate bounding box is filtered out if no horizontal line segment is detected inside of it. We employ the LSD (Line Segment Detector) algorithm proposed in [26] to detect horizontal line segments in a candidate bounding box. The LSD algorithm is robust in various illumination conditions, and exhibits good performance in detecting line segments on small-scale vehicles.

**Spatial-Temporal Memory Update:** A large variety of false candidates are removed by the semantic inference in 3D traffic and 2D image. But unfortunately, some true candidates are wrongly eliminated at the same time. To retrieve the missing detections and output coherent results, the spatial-temporal memory is used.

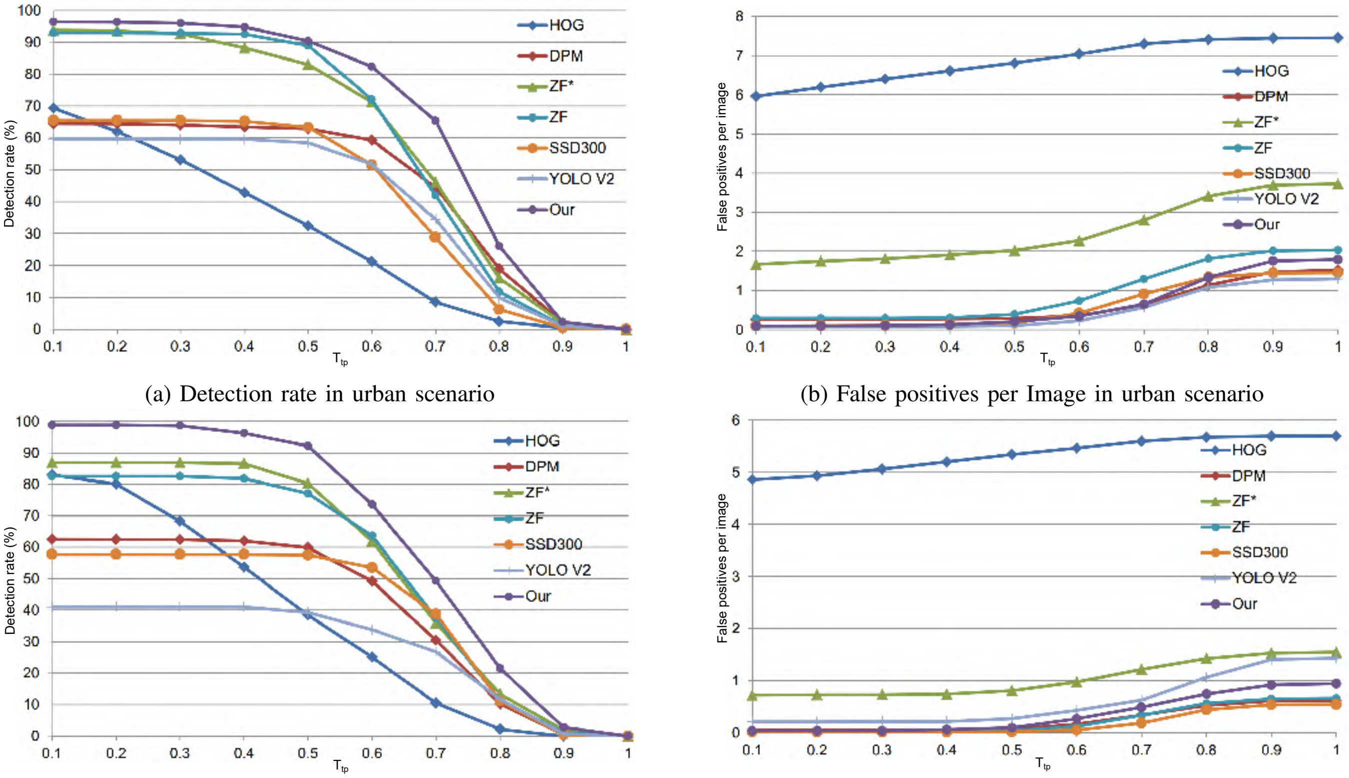
If the occasional missing detection occurs for a coherently detected vehicle which is with a high confidence, a virtual bounding box is reasoned by utilizing the detection results in previous frames in an interpolating manner. We estimate the confidence of a bounding box $b_n$ in $n_{th}$ frame by $N_n$ consecutive frames it steadily persists over. If $b_n$ is a steady bounding box, $N_n$ increases by 1, or $N_n$ decreases by $N_n/5$. If $N_n > 10$, $b_n$ is regarded as a bounding box with high confidence, and thus allowed a maximum missing detection number of $min(N_n/10, 10)$. The steadiness of $b_n$ is estimated by the location drifting of $b_n$ relative to $b_{n-1}$.

**Detection Output:** The detection output contains the information both in 2D image and 3D traffic. In 2D image plane, a vehicle is described by a bounding box $b = \{x_{tl}, y_{tl}, w, h\}$, where $(x_{tl}, y_{tl})$ denotes the top-left point of the bounding box, $w$ and $h$ denote the width and height of the bounding box respectively. In 3D traffic, a vehicle is described by the vehicle model $V = \{d_x, d_y, u_x, u_y, a_x, a_y, \beta, l_l\}$ defined in Eq. 3. $d_x$ and $d_y$ are obtained by mapping $b$ into the ego-vehicle coordinate using camera model $C$ in Eq. 4 that is resolved using the method in [11], $u_x = \dot{d_x}$, $u_y = \dot{d_y}$, $a_x = \dot{u_x}$, $a_y = \dot{u_y}$, $\beta = \arctan(u_y/u_x)$, and lane label $l_l$ is jointly resolved by $d_y$ and the lane markings that are detected using the method in [15].

## III. Experiments

### *A. Dataset and Setting*

**Dataset:** We collected a dataset in the real traffic scenes. The dataset includes 3,185 training images and 3,121 testing images. The training images are collected in the similar traffic scenes with that of the testing images. The testing images contain 1,821 images in urban scenario and 1,300



(a) Detection rate in urban scenario

(b) False positives per Image in urban scenario

(c) Detection rate in highway scenario

(d) False positives per image in highway scenario

Fig. 4: Performance of methods with different true positive threshold $T_{tp}$. SSD300 represents the model is trained using images that are resized to 300×300. To examine the influence of training data on the performance of method, we trained two ZF models, $ZF^*$ is trained on VOC [6] dataset and $ZF$ is trained on our own training images. Other models are also trained on our own training images.

images in highway scenario. The ground-truth boxes are annotated considering the nearest vehicles on the ego and two adjacent lanes.

**Evaluation Criterion:** Three metrics, detection rate, false positives per image and time consumption per image, are employed as evaluation criterion. For each detected box $B_d$, we compute its $IoU$ (Intersection of Union) with a ground-truth box $B_g$ by $IoU = (B_d \cap B_g)/(B_d \cup B_g)$. A true positive is counted if there exists a detected box $B_d$ whose $IoU$ exceeds the true positive threshold $T_{tp}$. If multiple detected boxes match with the same ground-truth box, only one true positive is counted. A detected box $B_d$ is counted as a false positive if its $IoU$ with all $B_d$ are smaller than $T_{tp}$. Detection rate and false positives per image rate is computed by the counted true positives and false positives.

**Comparative Methods:** Five methods, HOG-SVM [4], DPM-LSVM [7], Faster R-CNN [20], SSD [12], and YOLO V2 [19], are compared with our method. In the comparison with the Faster R-CNN, the ZF model is used.

### B. Result and Discussion

**Detection Rate:** As shown in Fig. 4a and Fig. 4c, our method achieves the higher detection rate. One major reason is that our method is independent of training data, thus exhibits the great advantage in the case of vehicles with extreme type. Vehicles with unusual appearance can hardly be detected by the learning-based methods if no similar vehicles is included in the training data. Another reason is our method presents good performance in detecting small-scale vehicles. The candidate generation procedure is conducted in 3D traffic where distant vehicles exhibit more information than in 2D image. Compared with our model, DPM, SSD300, and YOLO V2 achieve lower detection rate because they miss some small-scale vehicles and extreme-appearance vehicles.

DPM, SSD300, and YOLO V2 exhibit better performance on the urban scenario testing images than on highway testing images. The reason is that the highway scenario testing images contain a big truck with extreme appearance, and

| Methods | HOG | DPM | YOLO | SSD | $ZF^*$ | $ZF$ | Our |
|---|---|---|---|---|---|---|---|
| $DR^*$(%) | 32.5 | 62.9 | 58.4 | 63.4 | 83.0 | 89.0 | **90.5** |
| $DR^{**}$(%) | 38.4 | 59.9 | 39.3 | 57.6 | 80.3 | 77.1 | **92.2** |
| $FPPI^*$ | 6.81 | 0.29 | 0.10 | 0.17 | 2.02 | 0.39 | **0.21** |
| $FPPI^{**}$ | 5.34 | 0.07 | 0.27 | 0.02 | 0.81 | 0.04 | **0.10** |

TABLE I: Performance of methods on urban scenario testing images (*) and highway scenario testing images (**) when $T_{tp} = 0.5$. DR represents the detection rate, while FPPI represents false positive per image.

| **Methods** | HOG | DPM | YOLO | SSD | ZF | Our |
|---|---|---|---|---|---|---|
| **Time (C)** | 0.049 | 0.738 | – | – | 2.118 | **0.008** |
| **Time (G)** | – | – | 0.014 | 0.031 | 0.052 | – |

TABLE II: Mean time consumption per image of methods on the same CPU (C) and GPU (G). The unit is in seconds.

no similar vehicles are included in training data, causing a large number of missing detections, which indicate that the performance of learning-based methods is closely related with the training data. For the same reason, $ZF$ and $ZF^*$ model has the similar performance with our model on urban scenario testing images, but lower performance on highway scenario testing images. Since our training images are collected in the similar traffic scenes with that of the testing images, $ZF$ behaves better than $ZF^*$ on the urban scenario testing images. Some quantitative result can be found in Tab. I.

**False Positive Rate:** As shown in Fig. 4b and Fig. 4d, the DPM, $ZF$, SSD300, YOLO V2, and our method achieve lower false positive rate. The lower false positive rate of our method owes to not only the vehicle candidate generation mechanism in 3D traffic, but also the 2D/3D semantic and structural inference. The false detections with unreasonable scales are reduced by using the priori knowledge of the vehicle size in the actual 3D world. The false detections, which appear in unexpected locations like the sky, green belts and buildings, are avoided using the interdependence between vehicles and the road surface. Other false detections are further removed by using the vehicle interactions and vehicle appearance.

Compared with $ZF^*$ model, $ZF$ model greatly cuts down the number of false positives, which indicates once again that the performance of learning-based methods is closely related with the training data. Some quantitative result can be found in Tab. I.

**Time Consumption:** As shown in Tab. II, our method only needs 0.008s to process an image using an Intel i7 CPU. The high efficiency of our method benefits from the 3D road surface model that narrows down the candidate searching region and the fast line-scanning searching strategy that prevents the repeated local searching. We use the GPU of Nvidia GTX-1080.

## IV. CONCLUSIONS

This paper verifies the importance of semantic context and human common sense for vehicle detection by exploring a learning-free vehicle detection method. The proposed method outperforms other learning-based methods on a real traffic dataset, however, the method also presents some drawbacks. The method is based on extrinsic calibration parameters that are varying when the camera is shaking. In addition, the method is designed for a structured traffic scene, which is not generally applicable in some extreme scenarios like sharply curved roads. Therefore, in the future, we will focus on combing learning-based methods with the proposed idea to improve the accuracy and robustness of vehicle detection.

## REFERENCES

[1] A. Broggi, P. Cerri, and P. C. Antonello. Multi-resolution vehicle detection using artificial vision. In *IV*, 2004.

[2] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu. A new iterative triclass thresholding technique in image segmentation. *TIP*, 23(3):1038–1046, 2014.

[3] X. Clady, F. Collange, F. Jurie, and P. Martinet. Cars detection and tracking with a vision sensor. In *IV*, 2003.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Transactions on Graphics*, 31(4):1–10, 2012.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *T-PAMI*, 2010.

[8] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008.

[9] S. Y. Kim, S. Y. Oh, J. K. Kang, and Y. W. Ryu. Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion. In *IROS*, 2005.

[10] A. Kuehnle. Symmetry-based recognition of vehicle rears. *PR*, 1991.

[11] Q. Li, N. Zheng, and X. Zhang. Three-line calibration method for external parmeters of camera carried by car. *CN. Patent 1537749 A*, Oct. 20, 2004.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[13] H. Mori and N. M. Charkari. Shadow and rhythm as sign patterns of obstacle detection. In *International Symposium on Industrial Electronics*, 1993.

[14] A. Mukhtar, L. Xia, and T. B. Tang. Vehicle detection techniques for collision avoidance systems: A review. *T-ITS*, 2015.

[15] Z. Nan, P. Wei, L. Xu, and N. Zheng. Efficient lane boundary detection with spatial-temporal knowledge filtering. *Sensors*, 16(8):1276, 2016.

[16] J. Pan and T. Kanade. Coherent object detection with 3d geometric context from a single image. In *ICCV*, 2013.

[17] J. M. Park, H. C. Choi, and S. Y. Oh. Real-time vehicle detection in urban traffic using adaboost. In *IROS*, 2010.

[18] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi. Refinenet: Iterative refinement for accurate object localization. In *ITSC*, 2016.

[19] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

[21] M. Rezaei, M. Terauchi, and R. Klette. Robust vehicle detection and distance estimation under challenging lighting conditions. *T-ITS*, 2015.

[22] S. Sivaraman and M. M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *T-ITS*, 2013.

[23] B. Southall, M. Bansal, and J. Eledath. Real-time vehicle detection for highway driving. In *CVPR*, 2009.

[24] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *T-PAMI*, 2006.

[25] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[26] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *T-PAMI*.

[27] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization. *T-PAMI*, 2016.

[28] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *T-PAMI*, 2013.

[29] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5(6):495, 2004.

[30] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, 2016.