Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.016632

ARTICLE

Traffic Accident Detection Based on Deformable Frustum Proposal and Adaptive Space Segmentation

Peng Chen[1], Weiwei Zhang[1,*], Ziyao Xiao[1] and Yongxiang Tian[2]

[1]School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China

[2]Shanghai Fire Research Institute of MEM, Shanghai, 200032, China

[*]Corresponding Author: Weiwei Zhang. Email: zwwsues@163.com

Received: 13 March 2021 Accepted: 29 July 2021

ABSTRACT

Road accident detection plays an important role in abnormal scene reconstruction for Intelligent Transportation Systems and abnormal events warning for autonomous driving. This paper presents a novel 3D object detector and adaptive space partitioning algorithm to infer traffic accidents quantitatively. Using 2D region proposals in an RGB image, this method generates deformable frustums based on point cloud for each 2D region proposal and then frustum-wisely extracts features based on the farthest point sampling network (FPS-Net) and feature extraction network (FE-Net). Subsequently, the encoder-decoder network (ED-Net) implements 3D-oriented bounding box (OBB) regression. Meanwhile, the adaptive least square regression (ALSR) method is proposed to split 3D OBB. Finally, the reduced OBB intersection test is carried out to detect traffic accidents via separating surface theorem (SST). In the experiments of KITTI benchmark, our proposed 3D object detector outperforms other state-of-the-art methods. Meanwhile, collision detection algorithm achieves the satisfactory performance of 91.8% accuracy on our SHTA dataset.

KEYWORDS

Traffic accident detection; 3D object detection; deformable frustum proposal; adaptive space segmentation

1 Introduction

Vision-based object detection algorithms have been extensively exploited for traffic accident detection, which generates object location, motion information, and object category. However, accurate and robust traffic accident detection is difficult due to resolution constraints, illumination conditions, scale transform, and occlusion.

Recently vision-based researches on traffic accident detection [1,2] can achieve 2D bounding box regression and classification prediction from monocular images, and then utilize trajectory information to identify accidents. For example, Ijjina et al. [3] used Mask R-CNN [4] for object

detection followed by Centroid Tracking algorithm from RGB images, and then capitalized on speed and trajectory anomalies to infer traffic accidents. Unfortunately, these methods achieved poor accuracy and recall when objects are truncated, especially occluded. Besides, these works only provide texture information but lack depth information, which makes it difficult to describe objects for collision detection in 3D real-world scenes. As a result, 3D object detection method becomes the main research issue on traffic accident detection.

To address these limitations, this study develops a novel 3D object detector and adaptive space division model to quantitatively infer traffic accidents, as illustrated in Fig. 2. This method assumes the availability of 2D region proposals in RGB images, which can be easily obtained from some object detection frameworks [5–7]. Then, a set of deformable frustums are generated based on 2D region proposals and point cloud. Different from [8], this method extracts frustum-wisely features based on FPS-Net and FE-Net, and then uses a subsequent ED-Net to down-sample and up-sample these frustum-wisely features. Together with a detection header, the proposed method implements an end-to-end estimation of oriented 3D boxes. Subsequently, ALSR is developed to separate bounding boxes, and then SST is used to discriminate whether the reduced bounding boxes overlap, which can achieve collision detection. Extensive experiments are conducted on two datasets, KITTI benchmark [9] and our SHTA dataset. As illustrated in Fig. 1, SHTA dataset is collected in Shanghai urban roads by using surveillance cameras and LIDAR, which contains 5,672 crash records in different conditions.

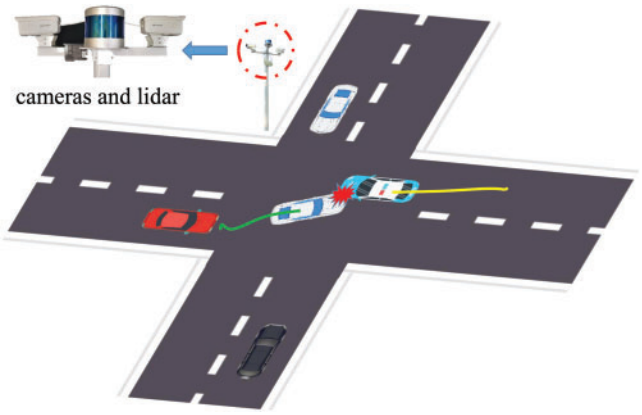

Figure 1: Recording platform of our SHTA dataset

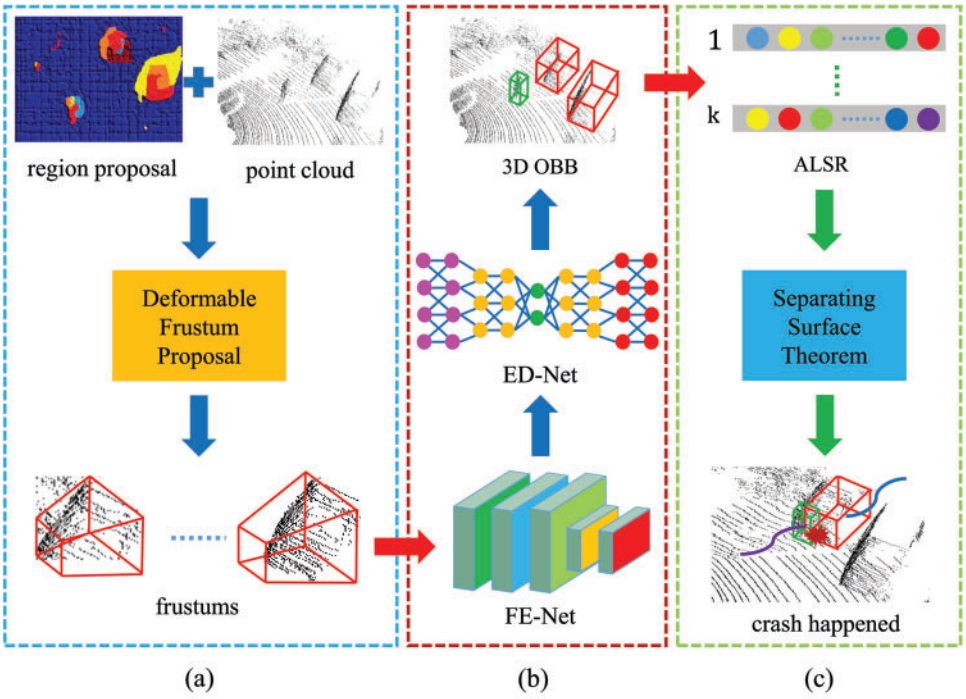

Figure 2: The pipeline of the proposed method. (a) Deformable frustum proposal based on 2D region proposals in an RGB image and point clouds. (b) 3D OBB regression framework. (c) Collision detection based on ALSR and SST

The contributions of this paper are summarized as follows:

(1) This method proposes a novel 3D object detector based on deformable frustum proposal, which can achieve an end-to-end 3D OBB regression in complex scenarios including occlusion and scale transform.
(2) This study proposes the adaptive least square regression model to split 3D OBB, which is followed by separating surface theorem to infer traffic accidents.

The rest of this study is organized as follows. Section 2 briefly reviews relevant researches on traffic accident detection, and some methods of 3D object detection based on RGB images and point clouds are summarized. The proposed method is fully presented in Section 3. Extensive experiments on the KITTI benchmark and SHTA dataset are elaborated in Section 4. Section 5 concludes our research of this paper.

2 Related Work

2.1 Traffic Accident Detection

Vision-based algorithms for traffic accident detection can be divided into two categories. One uses traditional image processing algorithms, such as optical flow. The other utilizes deep neural networks (DNN), such as Mask R-CNN. Yun et al. [10] predicted traffic accidents by using the motion interaction field (MIF), which uses the optical flow field and avoids complex vehicle tracking problems. Singh et al. [11] extracted deep representation via denoising autoencoders trained over the normal traffic videos, and then identified accidents based on the reconstruction

error and the likelihood of the deep representation. Ijjina et al. [3] utilized Mask R-CNN for object detection and centroid tracking algorithm, which can obtain speed and trajectory anomalies for further crash online inference. Chong et al. [12] used a convolutional LSTM auto-encoder to capture regular visual and motion patterns simultaneously for anomaly detection. Liu et al. [13] tackled the collision detection problem using the difference between a predicted future frame obtained by U-Net and its ground truth. In addition, Yao et al. [14] predicted traffic participants' trajectories and their future locations generated by an unsupervised deep learning framework to infer traffic accidents. Unfortunately, the precision and recall of the above methods are poor owing to resolution constraint, illumination condition, scale transform, and occlusion.

2.2 Multi-Sensor 3D Object Detection

Several image-based methods [15–19] focus on texture information but lack depth information, which increases the false detection rate of object detection algorithms. Besides, extensive point-based methods [20–23] provide depth information but ignore texture information. As a result, it could be helpful to integrate RGB images and point clouds to infer 3D OBB. MV3D [24] took the bird eye view (BEV) and front-view (FV) projection of point clouds together with corresponding RGB image as input, and then extracted feature using multi-stream CNNs for further 3D box regression. AVOD [25] introduced an early fusion architecture, which takes BEV and RGB images to generate high-resolution feature maps shared by two subnetworks: a region proposal network and a second stage detection network. MMF [26] proposed multiple related tasks for accurate multi-sensor 3D object detection including ground estimation and depth completion. F-PointNet [8] lifted 2D region proposals obtained by a 2D CNN detector to 3D frustum proposals, and then generated 3D OBB based on PointNet. F-ConvNet [27] slid frustum proposals to aggregate local point-level features with FCN for 3D OBB regression. However, the above methods belong to data-wise fusion, which is challenging owing to differences in data characteristics. Therefore, this method generates a specific frustum for each 2D region proposal and extracts frustum-wisely features via FPS-Net and FE-Net for 3D OBB regression.

3 The Proposed Method

As shown in Fig. 2, our system consists of two modules: 3D OBB regression and collision detection. Section 3.1 details some modules of 3D OBB regression framework, such as deformable frustum proposal, FE-Net, and ED-Net. Then, traffic crash is predicted using ALSR and SST in Section 3.2.

3.1 3D OBB Regression

By assuming the availability of 2D region proposals in RGB images that are easily obtained from some object detection frameworks, the proposed method generates a sequence of deformable frustums, and then extracts frustum-wisely features via FPS-Net and FE-Net. Subsequently, ED-Net is designed to down-sample and up-sample frustum-wisely features to obtain multi-scale feature maps. Together with a detection header, our method supports an end-to-end prediction of 3D OBB.

3.1.1 Deformable Frustum Proposal

To reduce search space and detect occluded objects, this method takes advantage of 2D region proposals in RGB images that are generated by region proposal network. Different from [8], our method assumes that the joint calibration of camera coordinate system and LIDAR coordinate system has already been performed, which means the optical axis of the camera is perpendicular

to each 2D region proposal. As illustrated in Fig. 3, a specific frustum is generated for each 2D region proposal by sliding along the optical axis between the image plane and the far plane. Owing to N ($N = 3$) aspect ratios of 2D region proposals generated by a 2D object detector, the width and height of frustums are changeable. In addition, our method uses a parameter σ ($\sigma = 0.9$, 1.0, 1.2) to control the depth of frustums, as the inaccurate depth between the image plane and the far plane is defined by the range of the depth sensor. Thus, each object owns deformable frustums $\{F_i = (W_i, H_i, D_i)\}_{i=1}^{N}$.

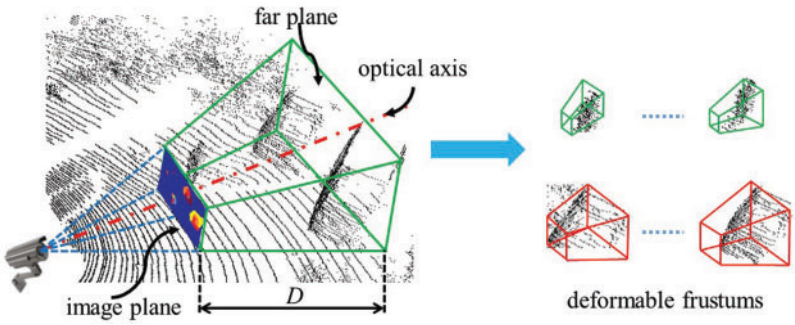

Figure 3: Illustration of how deformable frustums are generated for 2D region proposals in an RGB image

3.1.2 FE-Net

To extract frustum-wisely features, this method uses FPS-Net to obtain points $\{P_i = (x_i, y_i, z_i)\}_{i=1}^{n}$ from deformable frustums $\{F_i = (W_i, H_i, D_i)\}_{i=1}^{N}$ according to the principle of the most distant point. Then, our method applies FE-Net that includes convolution layers and max-pooling layers, followed by IR-Block, as shown in Fig. 4. Inspired by [28], IR-Block combines the Inception module with residual connection based on 1×1 convolution and 3×3 same convolution. Different from [27], the proposed method uses relative coordinates $\{\Delta P_i = (\Delta x_i, \Delta y_i, \Delta z_i)\}_{i=1}^{n}$ as the input of FE-Net. Each ΔP_i is obtained by subtracting each P_i with the centroid C of the frustum, $\Delta P_i = P_i - C$ for $i = 1, \ldots, n$.

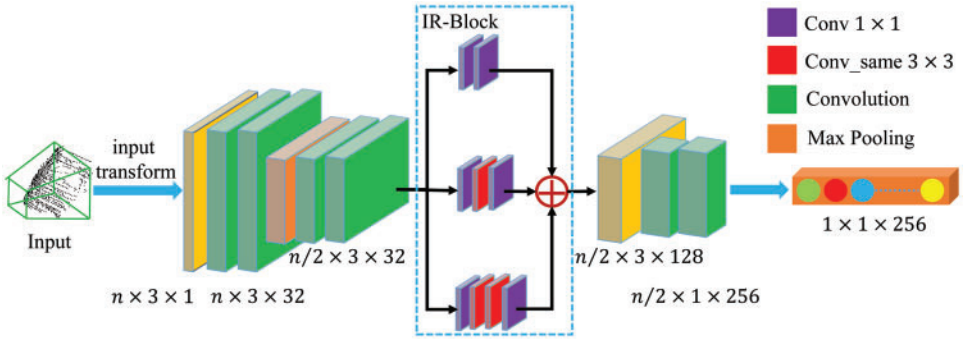

Figure 4: The architecture of FE-Net

3.1.3 ED-Net

To solve scale transform, the proposed method concatenates frustum-wisely feature vectors to form a feature map of the size $L \times d \times 1$ ($d = 256$, L is the number of objects), which is the input of subsequent ED-Net. As shown in Fig. 5, ED-Net is composed of two modules: (a) the encoder that gradually contracts the spatial dimension of feature maps with 1×1 convolution and 1×2 max pooling, and (b) the decoder that gradually expands the spatial dimension and object details based on 1×2 deconvolution. The size of the same convolution kernel is 3×3. Deconv layer upsamples the feature map to the same resolution as that in the encoder module, which is concatenated together along the feature dimension. On top of ED-Net is the detection header, which is used to classify object categories and generate 3D OBB. Given K categories, the classification branch outputs a feature map of the size $L \times (K+1)$. Our method uses focal loss [29] as the loss function, which helps solve the problem of data imbalance between different classes. Different from [8], this method parameterizes a 3D OBB by its center (x_c, y_c, z_c), shape (h, w, l), and yaw angle θ. The regression loss is smooth $L1$ loss and corner loss [8], which are used to regularize all parameters of a 3D OBB.

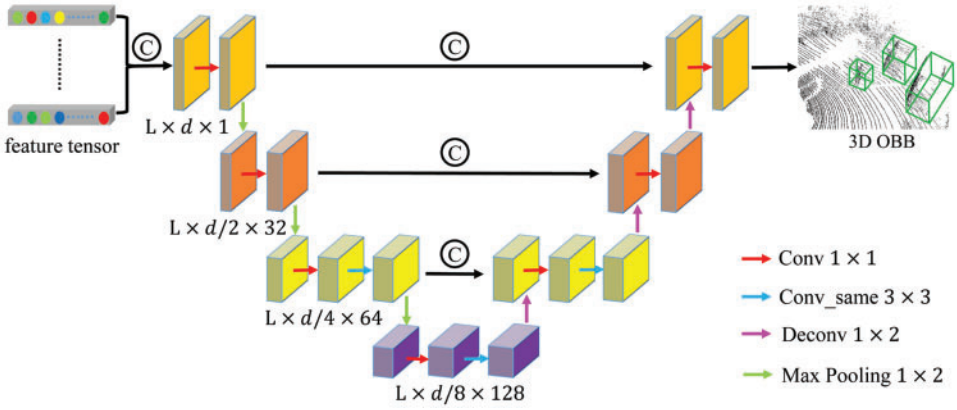

Figure 5: The architecture of ED-Net

3.2 Collision Detection

For collision detection, this method develops ALSR for 3D OBB segmentation and SST for reduced OBB intersection tests.

3.2.1 ALSR

To accurately infer traffic accidents, our method splits 3D OBB via ALSR. ALSR is a space partitioning method based on spectral clustering. First, ALSR uses an alternative optimization model to calculate the coefficient matrix. Then, the affine matrix is obtained based on the coefficient matrix. Finally, this method uses the standard segmentation method to split the affine matrix to obtain k subspaces.

The data matrix D of 3D OBB is reconstructed in directions of column and row, respectively. Thus, ALSR is established according to Eq. (1). C denotes column coefficient matrix and R means row coefficient matrix. λ is a regularization parameter, which is used to balance the impact of each other.

$$\min G(C, R) = ||D - DC - RD||_F^2 + \lambda(||C||_F^2 + ||R||_F^2) \tag{1}$$

To obtain C and R, this method uses an alternating optimization algorithm. First, C is fixed to optimize R, as shown in Eq. (2). Then, R is fixed to optimize C, as shown in Eq. (3).

$$R = D(I - C)D^{\mathrm{T}}(\lambda I + DD^{\mathrm{T}})^{-1} \tag{2}$$

$$C = (\lambda I + D^{\mathrm{T}}D)^{-1}D^{\mathrm{T}}(D - RD) \tag{3}$$

To speed up the convergence, this algorithm adopts the gradient descent method, as shown in Eqs. (4) and (5).

$$C^{(k+1)} = C^{(k)} - \rho_{kC}\nabla_C G(C, R) \tag{4}$$

$$R^{(k+1)} = R^{(k)} - \rho_{kR}\nabla_R G(C, R) \tag{5}$$

3.2.2 SST

SST is an iterative model to find a surface that can separate two 3D OBB. If the separating surface exists, they do not overlap. Fig. 6 shows that there are two bounding boxes A and B in two-dimensional space. X is the separation axis and Y is the separation line. T is the projection of the distance between C_a and C_b on the separation axis X. If $T > l_a + l_b$, they do not intersect. There are at most fifteen potential separating axes between a pair of 3D OBB. Due to ALSR generating k subspaces, this method makes at most $15 \times k$ adjustments.

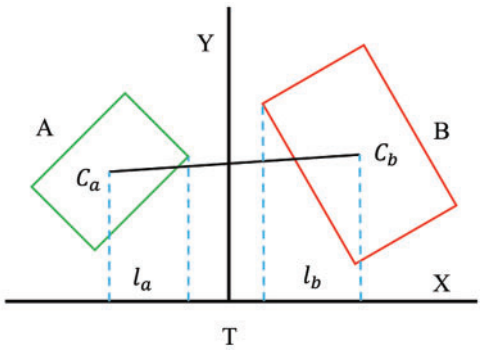

Figure 6: Separating surface theorem

4 Experiments

A large number of experiments are conducted on NVIDIA GTX 1080 GPU in this study. Section 4.1 depicts the KITTI detection benchmark and SHTA dataset. Section 4.2 evaluates the performance of 3D OBB regression, and then compares it with the state-of-the-art methods. The performance of collision detection is demonstrated in Section 4.3.

4.1 Datasets

4.1.1 KITTI

The KITTI detection benchmark contains 7,481 training and 7,518 testing RGB images and corresponding LIDAR point clouds, including three categories: Car, Pedestrian, and Cyclist. The proposed method trains our 3D object detector for Car and Pedestrian by splitting original data into 3,712 and 3,769 samples. The 3D IOU evaluation metrics are 0.7 and 0.5, respectively.

4.1.2 SHTA

To achieve detection of traffic accidents, our SHTA dataset was collected with surveillance cameras and LIDAR in Shanghai urban roads, including 5,672 crash records in different conditions such as occlusion and truncation. This method retrains our 3D object detector by fine-tuning the pre-trained network to generate 3D OBB. Then, traffic accident is predicted via ALSR and SST. As shown in Table 1, crash severity is classified into four levels, including no-injury, no-capacitating injury, incapacitating injury, and fatal injury.

Table 1: Division of traffic crash severity for our SHTA dataset

Traffic crash severity	Description	Frequency	Percent (%)
L1	No-injury	2,354	41.5
L2	No-capacitating injury	2,087	36.8
L3	Incapacitating injury	1,146	20.2
L4	Fatal injury	85	1.5
Overall	–	5,672	100.0

4.1.3 Implementation Details

Our 3D object detector are trained on KITTI benchmark and SHTA dataset, respectively. 3D object detector introduces 2D region proposals of Faster-RCNN [6] with ResNet-101 [30] for the KITTI detection benchmark. Using these 2D region proposals, this method samples 2,048 points from point clouds for each frustum based on FPS-Net. This method regards predicted boxes whose centers fall in the ground-truth boxes as positive samples and counts the others as negative samples. During both training and testing, data augmentation is used with random flipping (with a probability of 0.5) and random rotation (from $-\pi$ to π) to deal with angle transform. Before training, some hyperparameters need to be set. The batch size is set as 32 to generate 32 random samples of each frustum. The initial learning rate is 0.001 and decays one-tenth of the original every 30th epoch of the total 60 epochs. This method uses Adam optimizer with weight decay of 1e-05.

4.2 Evaluation of 3D OBB Regression

4.2.1 Comparing with SOTA

Table 2 illustrates the performance of our 3D OBB regression framework on the KITTI detection benchmark. This method outperforms most approaches in average precision (AP) and runtime. Especially, it improves upon the previously best model PV-RCNN [31] by 4.13 mAP, showing effectiveness of deformable frustum proposals. It gets better results on nearly all categories and has the biggest improvements on object categories that are often occluded (+1.37 AP for pedestrians and + 1.10 AP for cars). Besides, our method achieves great advantages in terms of speed (−0.03 s for runtime) thanks to the deformable frustum proposal decreasing the computing cost of point clouds. Fig. 7 shows the performance comparison of five methods (i.e., F-PointNet [8], F-ConvNet [27], PointRCNN [22], PV-RCNN [31]) including ours, under two different challenging conditions including scale transform and partial occlusion. Our method exceeds other methods under these conditions due to ED-Net providing multi-scale feature maps, texture information, and depth information. As a result, our 3D object detector is accurate and robust for traffic accident detection.

Table 2: 3D object detection AP (%) and runtime (s) on KITTI val set

Method	Runtime	Cars			Pedestrians			Cyclists		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [26]	0.36	71.09	62.35	55.12	–	–	–	–	–	–
VoxelNet [21]	0.5	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
F-PointNet [8]	0.17	81.2	70.29	62.19	51.21	**44.89**	40.23	71.96	56.77	50.39
AVOD [25]	0.08	76.39	66.47	60.23	36.1	27.86	25.76	57.19	42.08	38.29
PointRCNN [22]	0.1	89.96	75.64	70.7	47.98	39.37	36.01	74.96	58.82	52.53
F-ConvNet [27]	0.47	87.36	76.39	66.69	52.16	43.38	38.8	81.98	65.07	56.54
PV-RCNN31]	0.08	90.25	81.43	**76.82**	52.17	43.29	40.29	78.6	63.71	**57.65**
Ours	**0.05**	**92.13**	**82.53**	71.25	**53.54**	42.69	**40.59**	**83.96**	**66.49**	56.31

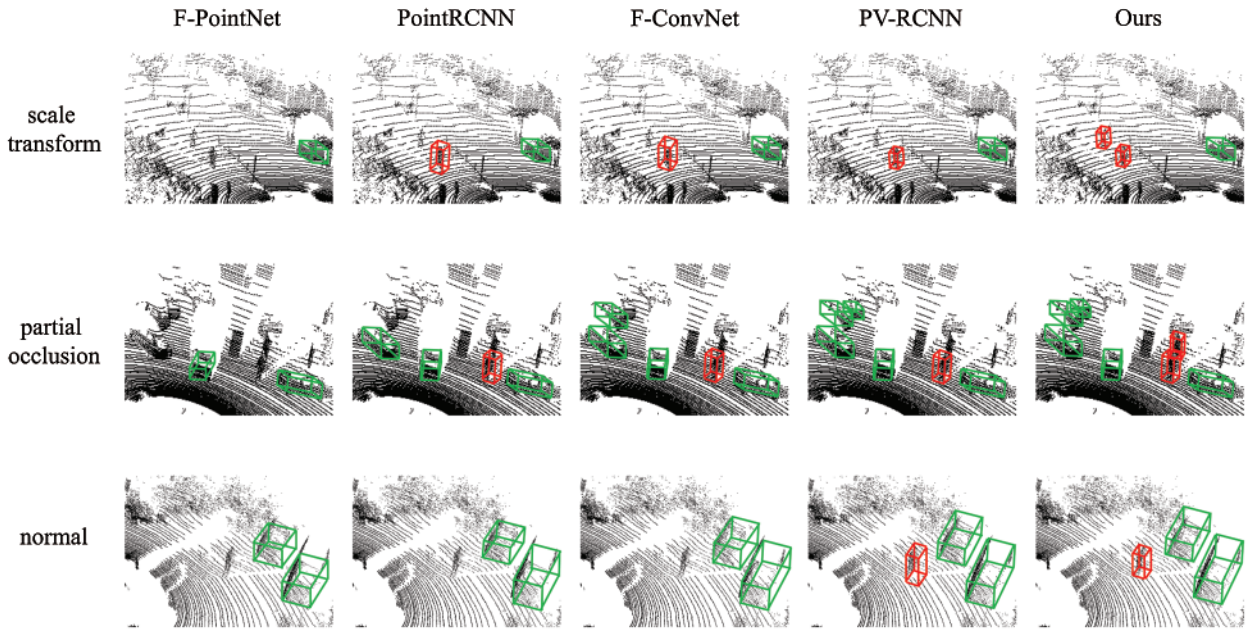

Figure 7: Qualitative results of five methods on the KITTI val set, under two challenging conditions including scale transform and partial occlusion. Green boxes denote car and red boxes represent predestrian

4.2.2 Effects of 2D Region Proposal

Using 2D region proposals in an RGB image, this method generates a sequence of deformable frustums based on point cloud. Thus, the performance of 2D region proposals plays an important role in 3D OBB regression. To investigate the influence of 2D region proposal on 3D OBB regression, six 2D object detectors are tested including RRC [32], MSCNN [33], MSCNN-ResNet50, MSCNN-ResNet101, FasterRCNN-ResNet50, and FasterRCNN-ResNet101. Table 3 shows the performance of 3D object detectors with different 2D region proposals. FasterRCNN-ResNet101 outperforms other methods on 2D region proposal owing to deeper backbone network, which makes traffic accident detection more precise.

Table 3: Effects of 2D region proposal on 3D object detection AP (%)

Method	Easy	Moderate	Hard
RRC	84.56	73.46	65.77
MSCNN	86.23	76.42	67.17
MSCNN-ResNet50	87.23	78.36	69.17
MSCNN-ResNet101	88.34	81.58	70.69
FasterRCNN-ResNet50	87.86	**83.31**	70.19
FasterRCNN-ResNet101	**89.98**	82.53	**71.25**

4.2.3 Effects of Frustum Feature Extractor

The proposed method extracts frustum-wisely features with FPS-Net and FE-Net. Thus, this method relies on the performance of the frustum feature extractor. Five approaches are compared to evaluate the effects of FE-Net on 3D OBB regression, including PointNet [34], PointNet++ [20], VoteNet [35] and PointCNN [36]. Table 4 illustrates the performance of 3D object detector with different frustum feature extractors. It is obvious that the performance of FE-Net is better than other methods in that IR-Block generates multi-scale frustum feature maps by combining the inception module with the residual connection.

Table 4: Effects of frustum feature extractor on 3D object detection AP (%)

Method	Easy	Moderate	Hard
PointNet	86.87	**83.46**	67.54
PointNet++	88.72	81.54	65.31
PointCNN	82.36	74.25	66.57
VoteNet	87.94	82.67	70.69
FE-Net	**89.98**	82.53	**71.25**

4.3 Performance of Collision Detection

4.3.1 Comparing with SOTA

The proposed method predicts traffic accidents with ALSR and SST. To assess the performance of collision detection, this method introduces three metrics including precision (P), recall (R) and frames per second (FPS) as shown in Eqs. (6) and (7).

$$P = \frac{\text{number of correct detection}}{\text{total number of detection}} \tag{6}$$

$$R = \frac{\text{number of correct detection}}{\text{total number of collision}} \tag{7}$$

Table 5 shows the comparison result of seven different collision detection approaches including Ijjina method [3], Yun method [10], Singh method [11], Chong method [12], Liu method [13] and Yao method [14]. Our method exceeds other approaches in terms of accuracy and runtime. Especially, it improves upon the previously best model by 3.24 P and 2.15 R thanks to adaptive space segmentation. In addition, it has the biggest improvements on the speed (+8 FPS), showing influence of separating surface theory.

Table 5: Collision detection P (%), R (%) and FPS on our SHTA dataset

Method	FPS	P	R
Yun method [10]	22	65.31	78.36
Singh method [11]	31	69.82	74.54
Ijjina method [3]	26	76.61	79.21
Chong method [12]	37	74.54	84.53
Liu method [13]	42	81.23	86.36
Yao method [14]	48	88.64	87.49
Ours	**56**	**91.88**	**89.64**

4.3.2 Influence of the Number of Subspaces

ALSR divides the 3D OBB into k subspaces. Then, SST makes $15 \times k$ judgments to infer traffic accidents. Thus, the number of subspaces plays an important role in collision detection. Table 6 illustrates the influence of k on the performance of collision detection. As k increases, the prediction accuracy (ACC) of our method firstly increases and then decreases. Experiments show ACC is better when k is 256.

Table 6: Effects of k on collision detection ACC (%)

k	4	16	64	256	512	1024	2048
ACC(%)	81.25	83.46	86.87	**91.88**	84.55	86.23	85.26

5 Conclusion

In this paper, a traffic accident detection system, based on 3D OBB regression and adaptive space segmentation, is proposed to effectively predict traffic accidents. A novel 3D object detector is adapted to generate 3D OBB for further traffic accident detection based on deformable frustum proposal, FE-Net, and ED-Net. Besides, this method proposes ALSR and SST for collision detection. Extensive experiments on SHTA dataset demonstrate this method outperforms other SOTA approaches in terms of accuracy and running speed. In the future, our SHTA dataset will be expanded and contain more different traffic scenes.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This work was supported in part by National Natural Science Foundation of China (No. 51805312); in part by Shanghai Sailing Program (No. 18YF1409400); in part by Training and Funding Program of Shanghai College young teachers (No. ZZGCD15102); in part by Scientific Research Project of Shanghai University of Engineering Science (No. 2016-19); in part by Science and Technology Commission of Shanghai Municipality (No. 19030501100); in part by the Shanghai University of Engineering Science Innovation Fund for Graduate Students (No. 18KY0613) and in part by National Key R&D Program of China (No. 2016YFC0802900).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Zu, H., Xie, Y. H., Ma, L., Fu, J. S. (2014). Vision-based real-time traffic accident detection. *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 1035–1038. Shenyang, China.
2. Elahi, M. M. L., Yasir, R. (2014). Computer vision based road traffic accident and anomaly detection in the context of Bangladesh. *International Conference on Informatics, Electronics & Vision*, pp. 1–6. Dhaka, Bangladesh.
3. Ijjina, E. P., Chand, D., Gupta, S., Goutham, K. (2019). Computer vision-based accident detection in traffic surveillance. *10th International Conference on Computing, Communication and Networking Technologies*, pp. 1–6. Kanpur, India.
4. He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969. Venice, Italy.
5. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448. Santiago, Chile.
6. Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems, 39,* 91–99. DOI 10.1109/TPAMI.2016.2577031.
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. et al. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision*, pp. 21–37. Amsterdam, The Netherlands.
8. Qi, C. R., Liu, W., Wu, C., Su, H., Guibas, L. J. (2018). Frustum pointnets for 3D object detection from RGB-D data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927. Salt Lake City, UT, USA.
9. Geiger, A., Lenz, P., Urtasun, R. (2012). Are we ready for autonomous driving? THE KITTI vision benchmark suite. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. Providence, Rhode Island.
10. Yun, K., Jeong, H., Yi, K. M., Kim, S. W., Choi, J. Y. (2014). Motion interaction field for accident detection in traffic surveillance video. *22nd International Conference on Pattern Recognition*, pp. 3062–3067. Stockholm, Sweden.
11. Singh, D., Mohan, C. K. (2018). Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems, 20(3),* 879–887. DOI 10.1109/TITS.2018.2835308.
12. Chong, Y. S., Tay, Y. H. (2017). Abnormal event detection in videos using spatiotemporal autoencoder. *International Symposium on Neural Networks*, pp. 189–196. Hokkaido, Japan.
13. Liu, W., Luo, W., Lian, D., Gao, S. (2018). Future frame prediction for anomaly detection–A new baseline. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545. Salt Lake City, UT, USA.
14. Yao, Y., Xu, M., Wang, Y., Crandall, D. J., Atkins, E. M. (2019). Unsupervised traffic accident detection in first-person videos. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 273–280. The Venetian Macau, Macau, China.
15. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J. (2017). 3D bounding box estimation using deep learning and geometry. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7074–7082. Honolulu, HI, USA.
16. Chabot, F., Chaouch, M., Rabarisoa, J., Teuliere, C., Chateau, T. (2017). Deep manta: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2040–2049. Honolulu, HI, USA.
17. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N. (2017). SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1521–1529. Venice, Italy.
18. Brazil, G., Liu, X. (2019). M3D-RPN: Monocular 3D region proposal network for object detection. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9287–9296. Long Beach, CA, USA.

19. Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F. et al. (2020). Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine, 37(4),* 34–49. DOI 10.1109/MSP.79.
20. Qi, C. R., Yi, L., Su, H., Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems, 30,* 5099–5108. DOI 10.3109/13816819409056905.
21. Zhou, Y., Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3D object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499. Salt Lake City, UT, USA.
22. Shi, S., Wang, X., Li, H. (2019). Pointrcnn: 3D object proposal generation and detection from point cloud. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–779. Long Beach, CA, USA.
23. Ali, W., Abdelkarim, S., Zidan, M., Zahran, M., El Sallab, A. (2018). YOLO3D: End-to-end real-time 3D oriented object bounding box detection from lidar point cloud. *Proceedings of the European Conference on Computer Vision*, Munich, Germany.
24. Chen, X., Ma, H., Wan, J., Li, B., Xia, T. (2017). Multi-view 3D object detection network for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915. Honolulu, HI, USA.
25. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S. L. (2018). Joint 3D proposal generation and object detection from view aggregation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–8. Madrid, Spain.
26. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R. (2019). Multi-task multi-sensor fusion for 3D object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7345–7353. Long Beach, CA, USA.
27. Wang, Z., Jia, K. (2019). Frustum convnet: sliding frustums to aggregate local point-wise features for amodal 3D object detection. *RSJ International Conference on Intelligent Robots and Systems*, pp. 1742–1749. The Venetian Macau, Macau, China.
28. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 125–135. San Francisco, California, USA.
29. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988. Venice, Italy.
30. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. Las Vegas, NV, USA.
31. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J. et al. (2020). Pv-rcnn: Point-voxel feature set abstraction for 3D object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538. Seattle, WA, USA.
32. Ren, J., Chen, X., Liu, J., Sun, W., Pang, J. et al. (2017). Accurate single stage detector using recurrent rolling convolution. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5420–5428. Honolulu, HI, USA.
33. Cai, Z., Fan, Q., Feris, R. S., Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. *European Conference on Computer Vision*, pp. 354–370. Amsterdam, The Netherlands.
34. Qi, C. R., Su, H., Mo, K., Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660. Honolulu, HI, USA.
35. Qi, C. R., Litany, O., He, K., Guibas, L. J. (2019). Deep hough voting for 3D object detection in point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286. Seoul, Korea.
36. Li, Y., Bu, R., Sun, M., Wu, W., Di, X. et al. (2018). Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems, 31,* 820–830. DOI 10.5555/3326943.3327020.