

Research Article

A 64-Line Lidar-Based Road Obstacle Sensing Algorithm for Intelligent Vehicles

Hai Wang ^{1,2} Xinyu Lou,¹ Yingfeng Cai ³ and Long Chen ³

¹Institute of Automotive Engineering, Jiangsu University, Zhenjiang 212013, China

²Robotic and Automation Lab, The University of Hong Kong, Pok Fu Lam, Hong Kong

³School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China

Correspondence should be addressed to Yingfeng Cai; caicaixiao0304@126.com

Received 16 August 2018; Accepted 4 November 2018; Published 21 November 2018

Guest Editor: Edward Rolando Núñez-Valdez

Copyright © 2018 Hai Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Based on the 64-line lidar sensor, an object detection and classification algorithm with both effectiveness and real time is proposed. Firstly, a multifeature and multilayer lidar points map is used to separate the road, obstacle, and suspension object. Then, obstacle grids are clustered by a grid-clustering algorithm with dynamic distance threshold. After that, by combining the motion state information of two adjacent frames, the clustering results are corrected. Finally, the SVM classifier is used to classify obstacles with clustered object position and attitude features. The good accuracy and real-time performance of the algorithm are proved by experiments, and it can meet the real-time requirements of the intelligent vehicles.

1. Introduction

Road target detection and classification is an important part of the safe driving of unmanned vehicles, especially in complex urban roads [1]. Among many kinds of sensors, due to the high resolution and precision of 64-line lidar, it is widely concerned by researchers and industrial developers. Unlike vision and millimeter radar, 64-line lidar has much bigger data amount to process which is more than 1 million 3D points per second at 10 Hz frequency, so that it has strict requirements on the real-time performance of the environment sensing algorithm.

There are two types of mainstream point cloud data processing methods. One is directly based on point cloud processing [2, 3], and the other is based on the grid map [4–6]. The former needs to process and classify every laser point which is extremely time consuming, while the latter one transfers 3D laser points to several 2D grid and then classify those new generated grids which is able to dramatically reduce classification calculation cost. The traditional grid map construction method contains several types such as mean height map [7], maximum height map, and minimum height map [8]. However, those methods that use

single threshold are difficult to divide obstacles of different heights and shapes [9, 10]. For example, they cannot distinguish between slopes, low obstacles, and roadside, and they often consider hanging objects as obstacles such as twigs above the vehicle height.

In laser points clustering, the computation cost of existing clustering algorithm such as K-means clustering, density clustering [11, 12], and hierarchical clustering [13, 14] are $O(m)$, $O(m^2)$ and $O(m^2 \log m)$ which directly relate to the point number m . The above clustering algorithm will increase the computation time, and it is difficult to meet the real-time requirements of unmanned vehicles.

Classification of obstacle targets around unmanned vehicles in dynamic environments is also very important for path planning and behavior prediction of the unmanned vehicle. In [15], the point clouds acquired by the lidar were projected onto the grid, clustered by the global nearest neighbor (GNN) and for each candidate, its eigenvector was calculated and classified by using the support vector machine (SVM) based on the radical basis function (RBF) kernel. In [16], the vehicles and pedestrians were classified using the Gaussian hybrid model classifier (GMM classifier). In [17], the point cloud is projected onto a 2D grid, and the

features of the envelope rectangular block in the grid map are extracted and classified by RPN network. In [18], the support vector machine is combined with the reflection intensity probability distribution, the longitudinal height contour distribution, and the position and attitude correlation features to classify the lidar point cloud features. The literature [19] combines the basic features of point cloud and contextual semantic environment to construct the original and extended feature vector of point cloud and use the support vector machine for target recognition.

In this work, an object detection and classification algorithm with both effectiveness and real time is proposed. The algorithm separates the road, obstacle, and suspension objects by using a multifeature and multilayer elevation map. Then, obstacle grids are clustered by a grid-clustering algorithm with dynamic distance threshold. After that, by combining the motion state information of two adjacent frames, the clustering results are corrected. Finally, the SVM classifier is used to classify obstacles with clustered object position and attitude features.

2. Grid Map Construction

This paper uses a multifeature multilevel height map to abstract lidar point cloud data. The multifeature multilevel height map is a variant based on multiscale height maps which divide the surrounding space of the vehicle into three layers. The first layer is the pavement layer, indicating the road surface where the vehicle is able to travel. The second layer is the obstacle layer, including various obstacles such as vehicles, pedestrians, buildings, traffic signs, trees, and so on. The last layer is the suspension object layer, indicating the obstacle whose height is greater than the vehicle's safe height and will not affect the vehicle's travel but is detected by the lidar. The algorithm flow is shown in Figure 1.

2.1. Grid Point Cloud Segmentation. The laser points located in the same grid are sorted according to the height from small to large so that a list of data points is obtained. Set the two-point interval height threshold as H_t . When the interval between the upper point and the lower point is greater than the interval height threshold, these two points belong to different plane blocks. Repeat the process to traverse the entire grid map to form all the plane blocks P_k . For each plane block that has several laser points in the grid, it contains five features: maximum height $\text{Max}H_k$, minimum height $\text{Min}H_k$, height mean $\text{Mean}H_k$, intensity mean $\text{Mean}I_k$, and intensity variance σ_k . Here, maximum height, minimum height, and height mean characterize the geometric characteristics of point clouds in plane blocks, and the other two reflect the reflection intensity characteristics of point clouds.

2.2. Pavement Layer Detection. Unlike most algorithms which only use height features for pavement layer segmentation, height and intensity information of point cloud are both used in this work.

2.2.1. Height Information. The maximum and minimum height of the plane block is used as the pavement classification feature. Due to the error of the lidar calibration, a two-step approach is used to judge the ground floor.

For each plane block H , $\Delta H = \text{Max}H - \text{Min}H$. If $\Delta H > b$, this plane block is considered as an obstacle. If $\Delta H < a$, this plane block is considered as a pavement plane. If $a < \Delta H < b$, intensity characteristics will be introduced because it is difficult to determine whether the plane block is an obstacle or a pavement by relying solely on the height feature. Here, a and b are threshold in which a will be smaller and b will be larger.

2.2.2. Intensity Information. When $a < \Delta H < b$, the block may be a road with a steep slope or an object with a small vertical height, so intensity information is needed for further judgment.

The intensity value of the lidar return is between 0 and 255. Here, we take a big amount of samples and count their intensity values. In this work, the lidar intensity value probability distribution curve of 200 vehicles, 200 pedestrians, and asphalt, cement pavement is obtained as shown in Figure 2. It can be seen that no matter a vehicle or a pedestrian, its intensity variance is bigger since its surface material and color are usually not uniform. On the other hand, the pavement property is relatively uniform, so its intensity distribution is relatively regular and the variance is small. Based on this, if the intensity variance of a block is less than the variance threshold $\text{Var}I_t$, the block will be taken as pavement plane otherwise it will be considered as obstacle.

2.3. Obstacle Layer and Suspension Layer Detection. After getting all the pavement layers, it is possible to obtain the average height H_G of all the pavement layers. Then, it is possible to set the suspension object layer height H_F as follows:

$$H_F = H_G + H_V + H_S, \quad (1)$$

where H_V is the height of the unmanned vehicle and H_S is the artificially set height of the obstacle from the roof of the vehicle while driving.

Hence, the plane block whose height is bigger than H_F will be considered as suspension layer while the rest is obstacle layer.

3. Obstacle Grid Clustering

Since we have projected the lidar point cloud into the grid map in the previous grid map construction step and obtained the obstacle grids, here the clustering time complexity reduced to $O(g)$, where g is the number of obstacle grids which is thousand times less than the number of raw lidar points.

Due to the fixed resolution of the lidar beam angle, its resolution will decrease as the distance increases which will lead to decompose a distant obstacle into multiple discrete parts and consider as multiple obstacles. To avoid this, we combine the obstacles motion state information of two adjacent frames to correct the spatial clustering results. The clustering algorithm flow chart is shown in Figure 3.

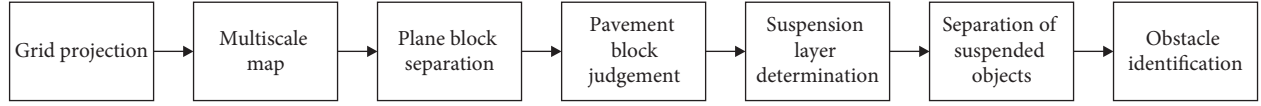


FIGURE 1: Flow diagram of multifeature multilayer height map construction.

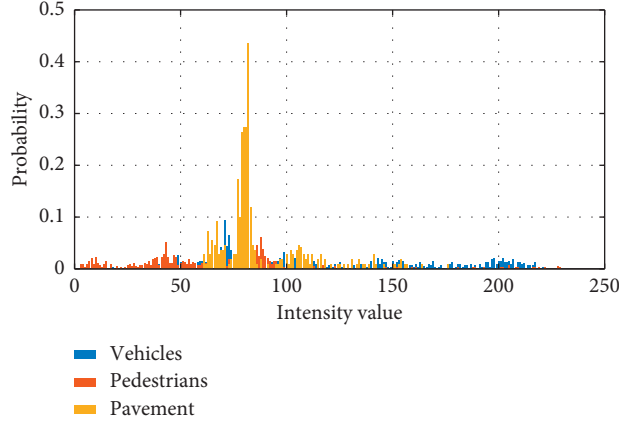


FIGURE 2: Probability distribution map of typical object return point intensity.

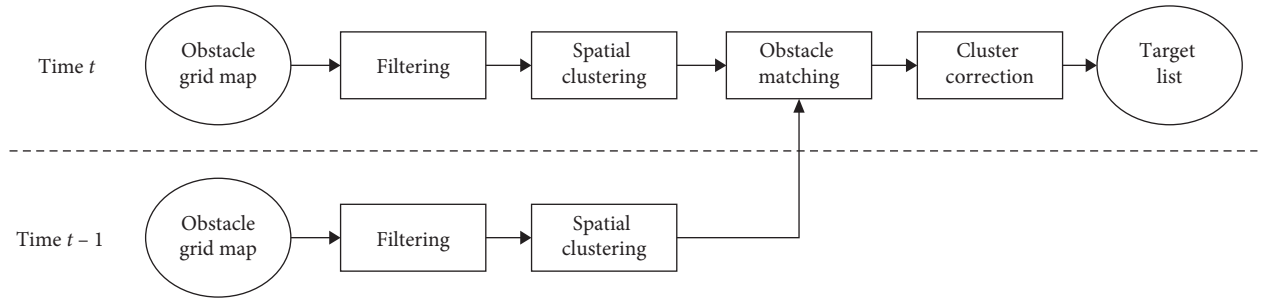


FIGURE 3: Clustering algorithm flow chart.

For different distance grid clustering in one frame, different distance thresholds are selected to cluster the discrete grids. The grid-clustering threshold is set as $N_T = D_T/G$, where G is the grid size and D_T is the radius parameter. For each grid, the adjacent grids in the area of $(2N_T - 1) \times (2N_T - 1)$ will be considered as one obstacle object.

In radius parameter setting, Borges proposed a method for calculating the radius parameter using distance values r_k [20], as shown in Figure 4.

Calculation formula of D_T is

$$D_T = r_{n-1} \frac{\sin(\Delta\varphi)}{\sin(20^\circ - \Delta\varphi)} + 3\sigma_r, \quad (2)$$

where r_{n-1} is the center coordinate distance of the obstacle grid, σ_r is lidar sensor measurement error, and $\Delta\varphi$ is horizontal angle resolution for lidar.

In order to further improve the accuracy of clustering, a target association matching clustering correction-based clustering is used. It pairs the closest obstacle blocks in the spatial clustering results at time $t-1$ and t by using four parameters: obstacle center coordinates, direction of motion, speed, and intensity mean.

4. Target Classification

In this work, dynamic obstacles in the road environment are separated into four categories: motor vehicles, nonmotor vehicles (bicycles), pedestrians, and others. Based on the motion characteristics and geometric contour characteristics of these four categories of obstacles, a SVM-based target classification method is proposed, as shown in Figure 5.

4.1. Target Feature. Since the information data of each obstacle are saved as a box model, the feature is also extracted from the box model. For each target box model, it has several groups of features listed as follows: (1) point X, point Y, point Z, and α which are the position and attitude features of the target; (2) length, width, height, and δ which are the contour features of the target. Here, α is the relative observation angle with the range of $(-\pi, \pi)$, as shown in Figure 6.

4.2. SVM Classifier. This work chooses the SVM classifier for obstacle classification which is good for small sample and nonlinear sample classification problems.

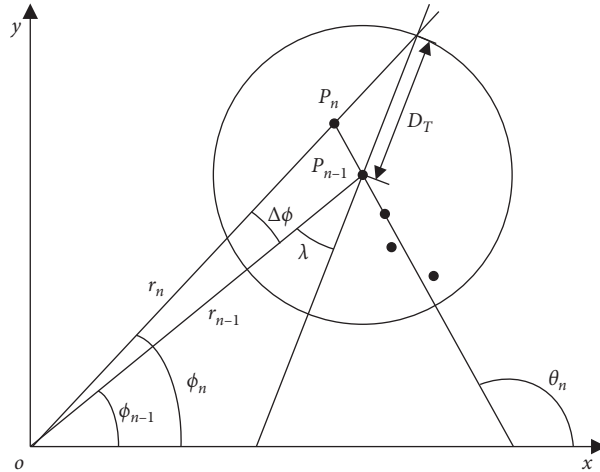


FIGURE 4: Radius parameter calculation method in [18].

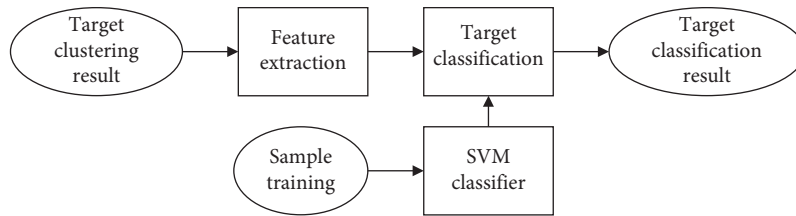


FIGURE 5: Classification process diagram.

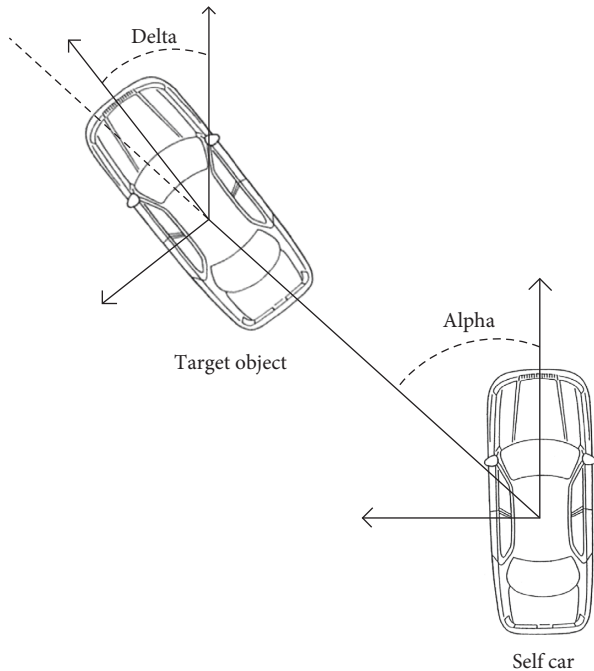


FIGURE 6: Schematic of *delta* and *alpha*.

In order to solve the nonlinear classification problem, the SVM classifier uses the kernel function to map the low-dimensional space classification problem to the

high-dimensional feature space to construct a linear function for classification. The radial basis kernel function (RBF) is used:

$$K(x, y) = \exp(-\gamma|x - y|^2). \quad (3)$$

5. Experiment and Analysis

5.1. Clustering Experiment and Analysis. The clustering algorithm is compared with the eight-connected clustering algorithm under fixed distance threshold and density-based spatial clustering of applications with noise (DBSCAN) algorithm. We use these three methods to carry out experiments on 200 target vehicles in the road environment. Through the experimental analysis, the clustering accuracy rate of the obstacle targets is shown in Table 1. Partition-based methods, such as k-means, need to know the number of clusters in advance, so they are not suitable for unmanned vehicles and we do not include them in the comparison. The average time taken by this algorithm is about 15 ms.

A set of intuitive comparisons are shown in Figure 7. Figure 7(a) shows the effect of the proposed clustering algorithm, Figure 7(b) is that of eight-connected clustering algorithm, and Figure 7(c) is that of DBSCAN. It can be seen that, in far distance, the eight-connected clustering algorithm marks one object incorrectly as multiple objects and the DBSCAN is a little better than that while our method still works well.

TABLE 1: Clustering algorithms comparison.

Target distance (m)	Eight-connected clustering algorithm accuracy (%)	DBSCAN accuracy (%)	Our method accuracy (%)
0–20	73.4	78.3	92.6
20–40	64.9	70.1	86.7
40–80	41.5	52.9	69.3
80–150	18.6	21.4	36.3

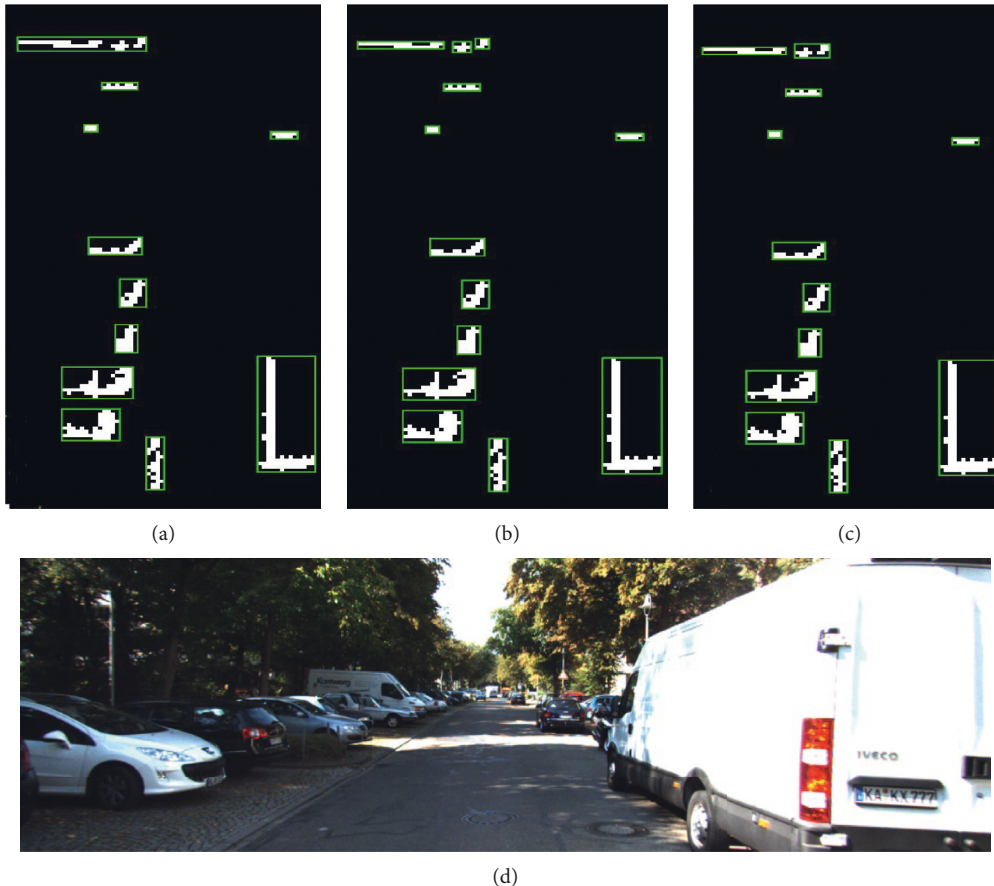


FIGURE 7: Clustering experiment. (a) Our method. (b) Eight-connected clustering. (c) DBSCAN. (d) Corresponding visual image.

5.2. Classification Experiment and Analysis. This experiment used the software developed by Professor Lin Chih-Jen of Taiwan University—LIBSVM [21]. In addition, the KITTI data set and the BDD100K are used for classification testing [22, 23]. Overall, there are 5217 samples containing 4091 vehicle samples, 417 bicycling samples, 573 pedestrian samples, and 136 other samples. Here, we take about 70% of the total number of samples as training samples and the rest as test samples. Through the grid optimization algorithm, the parameter penalty factor $C = 246$, the parameter $\gamma = 0.012065$, and the optimal recognition rate is 88.31%, as shown in Table 2.

A group of classification experiments is shown in Figure 8. Green, yellow, and red boxes mean vehicle, bicycle, and pedestrian, respectively, and the overall classification time is 10 ms per frame.

TABLE 2: Classification test result statistics.

	Overall	Vehicle	Bicycle	Pedestrians
Sample number	1566	1228	126	172
Correct classification number	1383	1172	71	138
Classification rate	88.31%	95.44%	56.35%	80.23%

6. Conclusion

Focusing on the difficulty of the large data of 64-line lidar which affect the real-time performance of unmanned vehicle, an object detection and classification algorithm with both effectiveness and real time is proposed. The algorithm separates the road, obstacle, and suspension by using a multifeature and multilayer elevation map. Then,

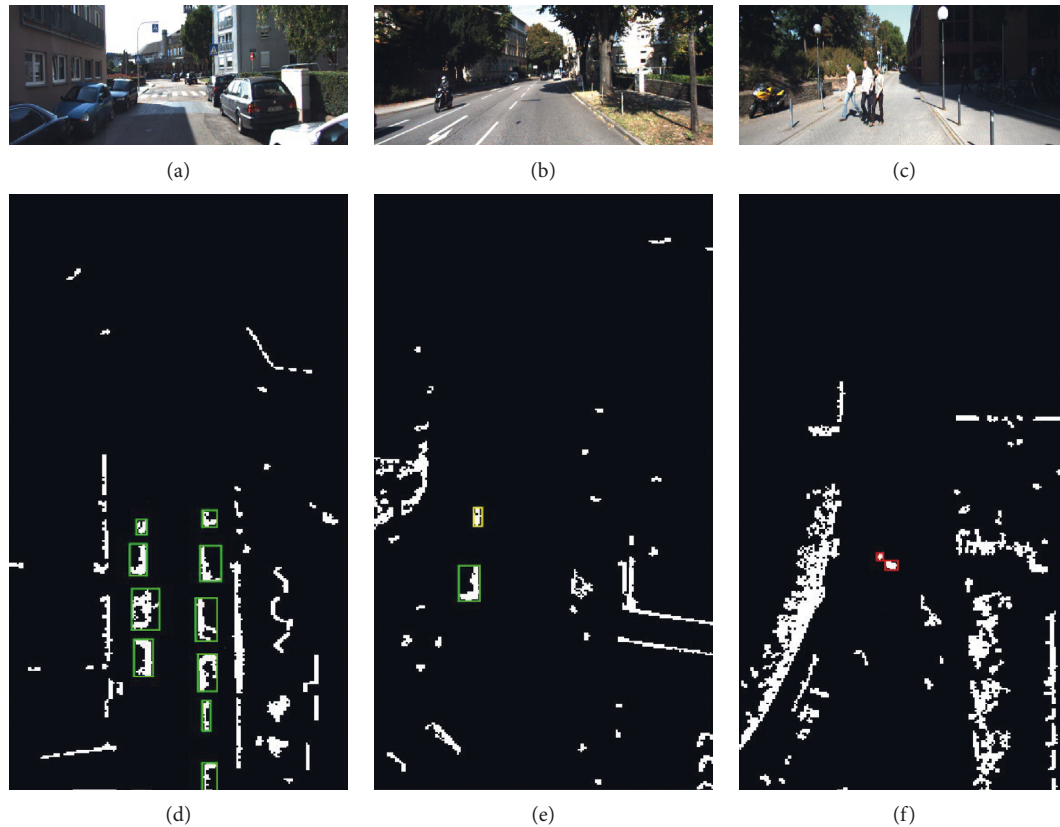


FIGURE 8: Target classification experiment result.

a grid-clustering algorithm based on dynamic distance threshold is used to cluster the obstacles, and the clustering results are corrected by combining the motion state information of two adjacent frames. Finally, SVM is used to classify obstacles. The experimental results show that the algorithm has good obstacle detection and classification accuracy and better real-time performance to meet the real-time requirements of unmanned autonomous vehicles while driving on the road. During the experiment, it was also found that the detection rate of bicycles and pedestrians is relatively low. This may be because the lidar can only scan a small part of pedestrians and bicycles far from the autonomous vehicle, and some of these parts are often filtered by the filtering algorithm or the features we use do not distinguish pedestrians and bicycles very well. So, in the future work, we will improve the filtering algorithm so that more obstacle information will be acquired and new features, such as speed, will be added to better distinguish pedestrians and bicycles.

Data Availability

The data sources in this work are from the public dataset KITTI.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

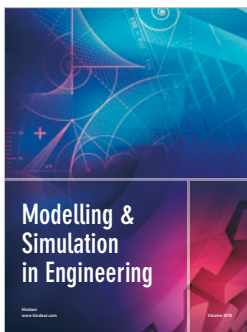
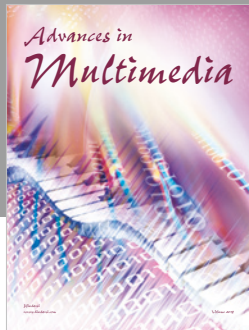
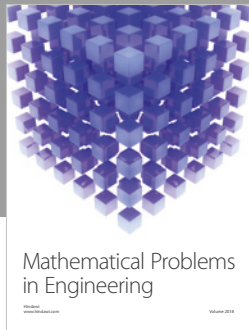
Acknowledgments

This study was supported by the National Key Research and Development Program of China (2018YFB0105003), National Natural Science Foundation of China (U1764264, 61601203, U1664258, U1764257, and 61773184), Natural Science Foundation of Jiangsu Province (BK20180100), Key Research and Development Program of Jiangsu Province (BE2016149), Key Project for the Development of Strategic Emerging Industries of Jiangsu Province (2016-1094 and 2015-1084), and Key Research and Development Program of Zhenjiang City (GY2017006).

References

- [1] Y. Cai, Z. Liu, H. Wang, and X. Sun, "Saliency-based pedestrian detection in far infrared images," *IEEE Access*, vol. 5, pp. 5013–5019, 2017.
- [2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77–85, Las Vegas, NV, USA, December 2016.
- [3] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.
- [4] S. Goga and S. Nedeveschi, "An approach for segmenting 3D LiDAR data using multi-volume grid structures," in

- Proceedings of Intelligent Computer Communication and Processing (ICCP)*, pp. 309–315, Cluj-Napoca, Romania, September 2017.
- [5] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nuchter, and J. Wang, “3D LIDAR point cloud based intersection recognition for autonomous driving,” in *Proceedings of 2012 IEEE Intelligent Vehicles Symposium (IV)*, pp. 456–461, IEEE, Madrid, Spain, June 2012.
- [6] Y. Liu, S. T. Monteiro, and E. Saber, “Vehicle detection from aerial color imagery and airborne LiDAR data,” in *Proceedings of 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1384–1387, IEEE, Beijing, China, July 2016.
- [7] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, “Multi-vehicle cooperative local mapping: a methodology based on occupancy grid map merging,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2089–2100, 2014.
- [8] S. Thrun, M. Montemerlo, H. Dahlkamp et al., “Stanley: the robot that won the DARPA grand challenge,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [9] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, “The obstacle detection and obstacle avoidance algorithm based on 2-d lidar,” in *Proceedings of 2015 IEEE International Conference on Information and Automation*, pp. 1648–1653, Lijiang, China, August 2015.
- [10] H. Wang, L. Dai, Y. Cai, X. Sun, and L. Chen, “Salient object detection based on multi-scale contrast,” *Neural Networks*, vol. 101, pp. 47–56, 2018.
- [11] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, “Fast density clustering strategies based on the k-means algorithm,” *Pattern Recognition*, vol. 71, pp. 375–386, 2017.
- [12] K. Yamazaki, “Effects of additional data on Bayesian clustering,” *Neural Networks*, vol. 94, pp. 86–95, 2017.
- [13] X. Zhang, H. Liu, and X. Zhang, “Novel density-based and hierarchical density-based clustering algorithms for uncertain data,” *Neural Networks*, vol. 93, pp. 240–255, 2017.
- [14] A. A. Liu, Y. T. Su, W. Z. Nie, and M. Kankanhalli, “Hierarchical clustering multi-task learning for joint human action grouping and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 102–114, 2017.
- [15] M. Darms, P. Rybski, and C. Urmson, “Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments,” in *Proceedings of 2008 IEEE Intelligent Vehicles Symposium*, pp. 1197–1202, Auburn Hills, MI, USA, February 2008.
- [16] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, “A lidar and vision-based approach for pedestrian and vehicle detection and tracking,” in *Proceedings of 2007 IEEE Intelligent Transportation Systems Conference*, pp. 1044–1049, Seattle, WA, USA, September 2007.
- [17] J. Behley, V. Steinhage, and A. Cremers, “Laser-based segment classification using a mixture of bag-of-words,” in *Proceedings of International Conference on Intelligent Robots & Systems*, pp. 4195–4200, Tokyo, Japan, November 2013.
- [18] P. Babahajiani, L. Fan, and M. Gabbouj, “Object recognition in 3D point cloud of urban street scene,” in *Proceedings of Asian Conference on Computer Vision*, pp. 177–190, Springer, Cham, Switzerland, November 2014.
- [19] S. Wirges, T. Fischer, J. B. Frias, and C. Stiller, “Object detection and classification in occupancy grid maps using deep convolutional networks,” 2018, <http://arxiv.org/abs/1805.08689>.
- [20] P. Skrzypczynski, “Building geometrical map of environment using IR range finder data,” *Intelligent Autonomous Systems*, pp. 408–412, 1995.
- [21] C. C. Chang and C. J. Lin, “LIBSVM: a library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.
- [22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: the KITTI dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [23] F. Yu, W. Xian, Y. Chen et al., *BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling*, arXiv preprint arXiv:1805.04687, 2018.




Hindawi

Submit your manuscripts at
www.hindawi.com

