

**ARTICLE INFO**

Article ID: 12-04-04-0031  
 © 2021 SAE International  
 doi:10.4271/12-04-04-0031

# Pedestrian Detection Method Based on Roadside Light Detection and Ranging

**Ziren Gong,<sup>1</sup> Zhangyu Wang,<sup>1</sup> Bin Zhou,<sup>1</sup> Wentao Liu,<sup>1</sup> and Pengfei Liu<sup>1</sup>**

<sup>1</sup>Beihang University, China

## Abstract

In recent years, to avoid the failure of the onboard perception system, intelligent vehicle infrastructure cooperative systems have been attracting attention in the field of autonomous vehicles. Using the perception technology of roadside sensors to provide supplementary traffic information for autonomous vehicles has become an increasing trend. Several roadside perception solutions select deep learning for three-dimensional (3D) object detection. However, deep learning methods have several issues and lack reliability in practical engineering applications. To tackle this challenge, this study proposes a pedestrian detection algorithm based on roadside Light Detection And Ranging (LiDAR) by combining traditional and deep learning algorithms. To meet real-time demand, Octree with region-of-interest (ROI) selection is introduced and improved to filter the background in each frame, which improves the clustering speed. Afterward, an improved Euclidean clustering algorithm was proposed by analyzing the scanning characteristics of LiDAR. Concretely, on account of the vertical and the horizontal angular resolution of the LiDAR, the authors propose a new method for determining the search radius of Euclidean clustering with adaptive distance. This algorithm can improve the problems of insufficient clustering of objects and the under-segmentation of neighboring objects. In addition, the authors' method carries out an improvement of the sampling mechanism of Pointnet++ to accomplish the classification, and the classification average precision (AP) of Pointnet++ for sparse point clouds is improved. The AP for pedestrian object detection can reach 94.67%, which is higher than that of the other two networks. What's more, the entire background filtering and clustering process takes 88.7 ms per frame, and the model obtained was deployed on NVIDIA Jetson AGX Xavier, attaining the inference time of 110 ms per frame, which can meet the speed requirement of LiDAR update and achieve the real-time application.

## History

Received: 11 Jun 2021  
 Revised: 17 Aug 2021  
 Accepted: 29 Oct 2021  
 e-Available: 12 Nov 2021

## Keywords

Infrastructure-based LiDAR,  
 Pedestrian detection,  
 Improved clustering,  
 Improved sampling  
 mechanism

## Citation

Gong, Z., Wang, Z., Zhou, B., Liu, W. et al., "Pedestrian Detection Method Based on Roadside Light Detection and Ranging," *SAE Int. J. of CAV* 4(4):413-422, 2021,  
 doi:10.4271/12-04-04-0031.

ISSN: 2574-0741  
 e-ISSN: 2574-075X



## Introduction

With the development of autonomous vehicles and intelligent vehicle infrastructure cooperative systems, environmental perception is becoming increasingly significant because of its proven effectiveness in traffic safety. However, in the future application of these technologies, the infrastructure that cooperates with these technologies is not yet developed. Roads and road users cannot collaborate. Pedestrians, as a vulnerable group of road users, are frequently harmed as a result of onboard sensor failures and obstruction of the driver's vision, as is evidenced by the recent fatal accident, in which a pedestrian was killed by a bus when he crossed the road in front of the bus [1].

To change the status quo of backward roadside infrastructure technology, a roadside perception scheme applied to the infrastructure is required, which can provide road users and traffic managers with more comprehensive real-time traffic information. Sensing solutions applied to roadside infrastructure must be equipped with reliable sensing technology. In previous studies, autonomous vehicles were equipped with Light Detection And Ranging (LiDAR), radar, cameras, GPS, etc. for environmental perception. To a certain extent, it improves the safety level of autonomous vehicles [2, 3, 4]. Owing to the wider detection range and the rich perceptual information of cameras and LiDAR, they are widely used in environmental perception. In a camera-based pedestrian detection system, a real-time pedestrian detection system is often designed by using Convolutional Neural Network (CNN) for autonomous vehicles [5, 6]. In recent LiDAR-based pedestrian detection research, deep learning is the most commonly used method, such as DA-ANN [7], Semantic Voxels [8], and three-dimensional (3D)-CNN [9]. In this study, we propose a LiDAR-only pedestrian detection method.

There are three reasons for choosing LiDAR as the roadside sensing device in this study: (1) LiDAR is less affected by lighting conditions, and it can function well at night; (2) LiDAR can provide the location and distance information of the object in real time, which is difficult for the camera; (3) Despite the fact that a wider detection range and depth information can be obtained by using more cameras, a single LiDAR can accomplish the above tasks and save more computing power.

For LiDAR-based pedestrian detection methods, although 3D object detection networks can be directly deployed to the roadside, the deep learning method is sometimes unreliable in engineering applications: there may be missed objects or incorrect classifications [10]. We reveal the intrinsic reasons from the following three aspects, (1) Limited by the uneven density of point clouds, fewer valid points are scanned on the pedestrian. (2) Various background objects, such as trees, bushes, and poles, might be close to the pedestrians, which results in error detection. (3) Noising points in the dataset caused by equipment and scanning characteristics (like the LiDAR we used) interfere with the results of deep learning. Combining deep learning methods with the traditional method can solve the above problems and improve the

reliability of pedestrian detection to a certain extent, and it is also a common method in engineering applications.

To overcome the above problems of deep learning, the following methods are used to enhance the reliability of pedestrian detection:

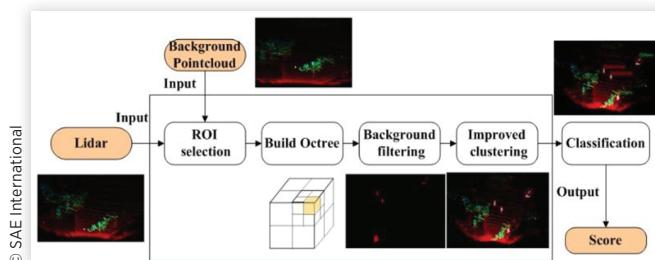
- Filter out background objects that interfere with pedestrian detection.
- Improve the learning ability of the classification network for sparse point clouds.

The pedestrian detection algorithm, which combines traditional and deep learning methods, generally consists of three parts: background filtering, clustering, and classification. Background filtering is often the first step in the preprocessing of the original point cloud. By filtering out the background, such as trees, buildings, and the ground, the accuracy of clustering and classification is ensured. Common background filtering algorithms include a two-dimensional grid-based method [11], a background filtering algorithm based on height information [12], and a density-based background filtering algorithm [13].

Clustering algorithms are relatively mature, mainly including density clustering [14], K-means clustering [15], and hierarchical clustering [16]. Clustering based on density, as illustrated by density-based spatial clustering of applications with noising (DBSCAN), is primarily performed by characterizing the density of the sample distribution. Partitioned clustering, such as K-means clustering, is performed by dividing the point cloud into several clusters. Balanced iterative reducing and clustering using hierarchies (BIRCH) is a clustering algorithm that belongs to hierarchical clustering, which is performed by dividing the clustering space in which clustering results are combined to complete the clustering. When selecting a clustering algorithm for environmental perception, the real-time generalization and effectiveness of the algorithm should be considered comprehensively.

Generally, classification algorithms are primarily divided into methods based on the bird's-eye view (BEV) and point-wise features. The methods based on the BEV [17, 18] include three parts: BEV generator, network backbone, and detection head. The BEV feature map is generated by the BEV generator as a feature and input into the network. The detection head outputs the size, location, and object category of the bounding box. Point-wise methods [19, 20, 21] usually include three parts: LiDAR representation, network, and detection head. This method directly uses the original point cloud as the input to the network and extracts the point features in the network backbone. Similar to the BEV-based method, classification and positioning tasks are completed through the detection head.

Although previous studies in background filtering, clustering, and classification have been relatively mature, there is still room for improvement in each method. In terms of background filtering; although the aforementioned methods have their own advantages in accuracy, speed, and stability, it is difficult to consider all of these factors at the same time. The same problem also exists in clustering. It is necessary to

**FIGURE 1** Structure of the proposed method.

comprehensively consider the characteristics of LiDAR, segmentation effect, and real-time performance [22]. The K-means algorithm needs to specify the number of clusters in advance, which is difficult to adapt to the number of dynamically changing objects [23]. DBSCAN has a poor clustering effect on data with uneven densities [24]. For classification, the classification effect is frequently affected by the uneven characteristics of the point cloud.

To solve these problems, solutions and algorithms are explored in this study, and a pedestrian detection method based on roadside LiDAR is proposed. As shown in Figure 1, the pedestrian detection method used in this study consists of the following three parts:

- Background filtering: Filter the ground and static obstacles (such as trees, chairs, etc.) in each frame to retain dynamic objects.
- Clustering: Cluster point clouds into multi-objects with improved Euclidean clustering.
- Classification: Use an improved Pointnet++ network to classify the clusters and identify pedestrian and non-pedestrian objects.

For background filtering, Octree is used with region-of-interest (ROI) area selection to filter the background, such as the ground, trees, and trash cans, which improves the speed and accuracy of clustering. This background filtering algorithm can filter more than 97.6% of the background points. In clustering, this study analyzes the scanning characteristics of LiDAR at vertical and horizontal angles and proposes an adaptive distance Euclidean clustering algorithm, which improves the problem of insufficient clustering or under-segmentation of neighboring objects. This process took 88.7 ms. For classification, this study selected Pointnet++ as the network [21], which is suitable for clusters. In addition, according to the characteristics of the sparse point cloud, the local area sampling mechanism of Pointnet++ was changed to improve the average precision (AP) of the network. The entire algorithm is suitable for sparse point clouds. The experimental results show that the roadside pedestrian detection method proposed in this study runs 94.67% in AP, which is better than the same type of PointRCNN and BEV-based Voxelnet. This study summarizes the following innovations.

- A Euclidean clustering algorithm with adaptive distance is proposed, which can improve the problems of

insufficient clustering and under-segmentation of neighboring objects.

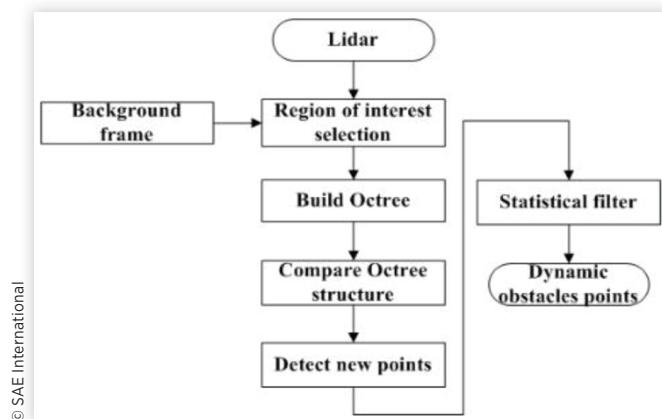
- The local area sampling mechanism of Pointnet++ was improved, increasing the AP of Pointnet++ to 94.67% on roadside data.

## Methodology

### Background Filtering

To avoid the interference of the background point cloud on clustering, it is required to only keep the point cloud of road users, such as vehicles, pedestrians, and bicycles. This study accomplishes this task through background filtering in two steps. One is that the roadside scene remains unchanged, and the other is that this processing method can improve the speed of the entire traditional algorithm process. In previous studies, it has been proven that Octree has advantages in processing point clouds [25, 26]; therefore, this study selects Octree for background filtering.

As shown in Figure 2, the filtering algorithm in this study includes ROI area selection, building Octree, comparing leaf nodes, and outlier filtering. The ROI area selection in this process must be explained separately. This step filters the point cloud outside the road and retains the point cloud area of interest, where the road users are located. Practice has shown that after background filtering, the speed of the entire traditional algorithm, including background filtering and clustering, can reach 88.7 ms per frame, and in most cases, the background filtering can filter more than 97.6% of the background points. Table 1 presents a detailed evaluation of the algorithm with the number of the objects' points and background points before and after the process. The evaluation index we used is the Filtering percentage used in [13]. As pedestrian detection is our task in this study, we divided the task into three scenarios during evaluation: pedestrian-only, pedestrian-cyclist, and pedestrian-cyclist-vehicle. As shown

**FIGURE 2** Background filtering process.

**TABLE 1** Evaluation of background filtering at different scenarios.

	Pedestrian-only	Pedestrian-cyclist	Pedestrian-cyclist-vehicle
Background points before	21,798	21,708	21,507
Background points after	349	456	511
<b>Filtering percentage</b>	<b>98.4%</b>	<b>97.9%</b>	<b>97.6%</b>
Pedestrian points before	103	92	93
Pedestrian points after	99	89	91
Pedestrian excluded percentage	4.9%	3.3%	2.2%
Cyclist points before	NA	51	59
Cyclist points after	NA	48	57
Cyclist excluded percentage	NA	5.9%	3.4%
Vehicle points before	NA	NA	262
Vehicle points after	NA	NA	255
Vehicle excluded percentage	NA	NA	2.7%

© SAE International

in Table 1, the algorithm keeps most features as well as LiDAR points of objects. Though there were still some noising points after the process, the outlier filter would clear them all.

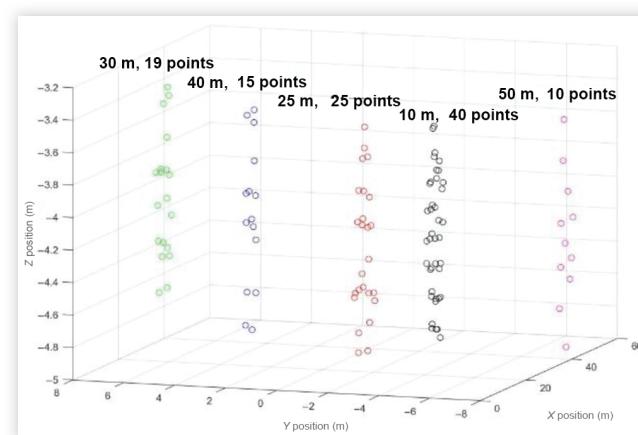
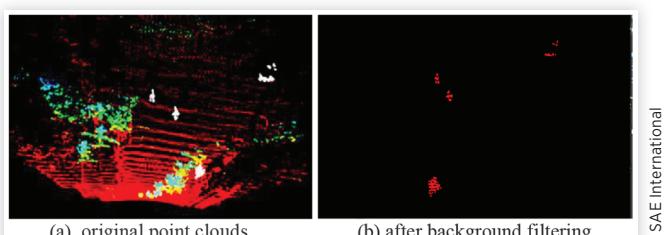
In Figure 3, our algorithm shows a good performance in filtering the background points and keeping objects' points [Figure 3(a) and (b)].

## Clustering

After background filtering, only the point cloud of the road users remains, and these points require to be clustered. Owing to the uneven density of point cloud, it is often “near dense and far sparse,” and the change of density with distance will cause clustering errors [27]. In this study, the number of point clouds at different distances was counted, as shown in Figure 4. When a pedestrian is 10 m away from the LiDAR, the point cloud can reach 40 points. When a pedestrian is 50 m away from the LiDAR, the point cloud has only 10 points, and the point cloud is sparse.

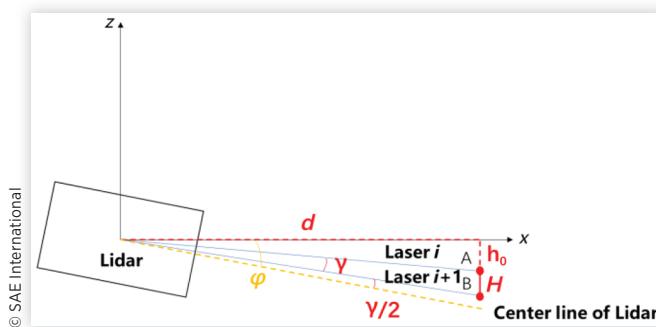
The effect of the original Euclidean clustering depends on the setting of the initial threshold, and there are often problems that distant objects cannot be fully clustered and near objects are under-segmented. To address these problems,

this study implements an adaptive distance Euclidean clustering algorithm, which selects an appropriate threshold for Euclidean clustering based on the distance between the point cloud and LiDAR. In addition, the choice of threshold is related to the accuracy of the LiDAR. Considering the structure and characteristics of LiDAR, this study analyzes the scanning characteristics of LiDAR in the vertical and horizontal directions. The minimum search radius  $H$  in the vertical direction is related to the vertical angular resolution and the distance of the point cloud from the LiDAR on the X-Y plane. As shown in Figure 5,  $d$  is the distance between points A and B on the X-Y plane to the LiDAR, the vertical distance between adjacent points A and B in the Z-axis direction  $H$  is the minimum distance in the same cluster,  $h_0$  is the distance from point A to the horizontal line,  $\varphi$  is the inclination angle when the LiDAR is installed, and  $\gamma$  is the vertical angular resolution of the LiDAR. The vertical distance  $H$  between adjacent points A and B in the same cluster can be expressed by Equations 1 and 2. Therefore, the Euclidean

**FIGURE 4** Original point cloud and after background filtering.**FIGURE 3** Original point clouds and after background filtering.

© SAE International

**FIGURE 5** Vertical distance  $H$  of points A and B detected by adjacent lasers at distance  $d$ .



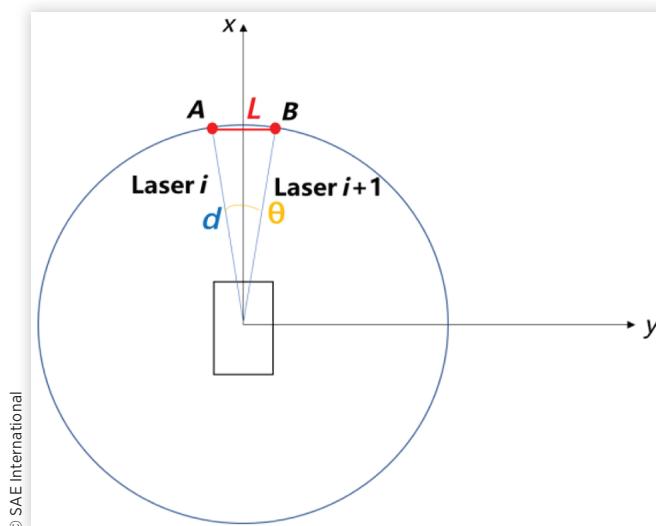
clustering threshold at distance  $d$  must first meet the minimum search radius  $H$  in the vertical direction.

$$H = d \cdot \tan\left(\varphi - \frac{1}{2}\gamma\right) - h_0 \quad \text{Eq. (1)}$$

$$h_0 = \tan\left(\varphi - \frac{3}{2}\gamma\right) \quad \text{Eq. (2)}$$

The minimum search radius  $L$  in the horizontal direction is related to the horizontal angular resolution and the distance of the point cloud from the LiDAR on the X-Y plane. As shown in Figure 6, the horizontal distance  $L$  between adjacent points A and B is the minimum search radius required to gather two points in the same cluster, and  $\theta$  is the horizontal angular resolution of the LiDAR because distance  $d$  from the point to the LiDAR is much larger than the horizontal distance  $L$  between two adjacent points. Therefore,  $L$  can be approximated as the arc length between two adjacent points. As shown in Equation 3, the threshold of Euclidean clustering in the

**FIGURE 6** Horizontal distance  $L$  between points A and B detected by adjacent lasers at distance  $d$ .



horizontal direction should meet the minimum distance  $L$  between adjacent points as follows:

$$L = d \cdot \frac{\theta \cdot \pi}{180} \quad \text{Eq. (3)}$$

As analyzed above, the selection of the threshold should meet the detection characteristics of LiDAR simultaneously in the vertical and horizontal directions. Moreover, the threshold should change with distance. In this study, the threshold is set as shown in Equation 4. The minimum search radius  $r$  includes the minimum search radius  $H$  and  $L$  of the two points in the vertical and horizontal directions, respectively.  $\beta$  represents an adjustment parameter related to the accuracy of LiDAR. The higher the accuracy of LiDAR the lower the value of  $\beta$ , and vice versa.  $\alpha$  represents the minimum tolerance distance, which ensures the overall integrity of the near-point cloud segmentation. After Euclidean clustering with an adaptive threshold, the point cloud is clustered into its own object.

$$r = \beta \sqrt{H^2 + L^2} + \alpha \quad \text{Eq. (4)}$$

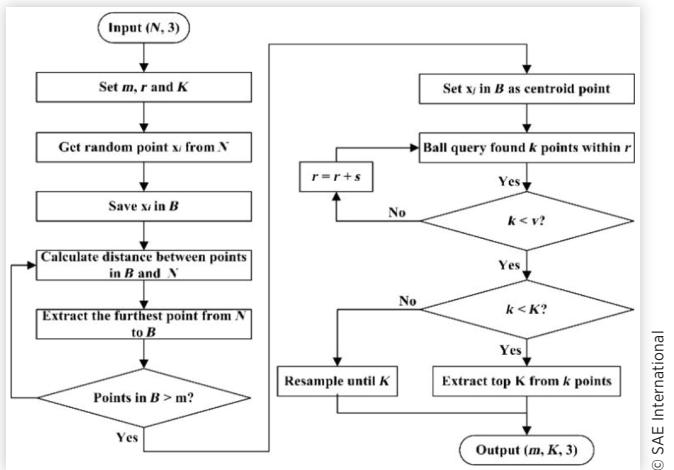
## Classification

Pointnet++ uses the set abstraction (SA) layer for hierarchical local feature sampling and the farthest point sampling algorithm for uniform downsampling in the sampling layer. It is assumed that the input point cloud is  $N = \{x_1, x_2, \dots, x_n\}$ ; after the process in Table 2,  $m$  points are selected as the center points in the grouping layer. The sampling layer downsampling the point cloud is more uniform than random sampling, and it can cover all points in the space as much as possible.

The grouping layer uses a ball query on  $m$  central points to obtain a fixed-length local area feature vector. When performing ball-query sampling, the point cloud within the radius is sampled using a preset threshold. However, owing to the volatility and inhomogeneity of the point cloud, there are often isolated points in distant objects. When the farthest point sampling selects these points as the center point, because

**TABLE 2** Process of sampling layer.

Step 1. Set the number of sampling points $m$ .
Step 2. Randomly select a point $x_i$ from the input point set $N$ and store it in the center point set $B = \{x_i\}$ .
Step 3. Select the remaining $n - 1$ points to calculate the distance to point $x_i$ , and select the farthest point $x_j$ to write in the starting point set $B = \{x_i, x_j\}$ .
Step 4. Select the remaining $n - 2$ points to calculate the distance to each point in set $B$ , and use the shortest distance as the distance from the point to the point set so that the $n - 2$ distances to the point set are obtained, and the farthest one is selected to write in starting point set $B = \{x_i, x_j, x_k\}$ .
Step 5. Repeat the above steps until the number of center points reaches $m$ .

**FIGURE 7** Process of improved SA layer.

## Experiment

To verify the advancement of the algorithm in this study, in the pedestrian detection effect experiment, the algorithm was compared with two other pedestrian detection algorithms. They are representative networks based on BEV, Voxelnet, and a representative network based on point-wise PointRCNN [20]. In this experiment, by comparing this algorithm with the original Pointnet++ vertically, it is verified whether the improved sampling mechanism is effective. A horizontal comparison of the precision of Voxelnet and PointRCNN on the experimental data proves the advancement of the algorithm.

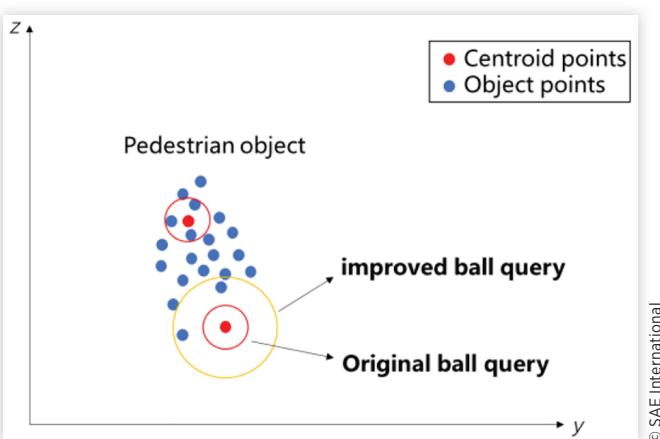
In addition, this study also identified that in the improved sampling mechanism, different step sizes  $s$  and threshold  $v$  will affect the precision of Pointnet++. Therefore, in the parameter optimization experiment, the best step  $s$  and threshold  $v$ , which could ensure that the network performs its best, were identified.

## Roadside Equipment Installation

LiDAR is a sensor that collects the size and distance information of the surrounding environment through emitted lasers. It is not easily affected by light or weather conditions. It can work both during day and night and is suitable for outdoor road traffic information collection [29]. In this study, a 64-line Livox Horizon is used to perceive obstacles and the environment in an intersection scene. The LiDAR has a detection range of up to 260 m, a horizontal field of view of 81.7°, and a vertical field of view of 25.1°. The collected point cloud data include the position information of each point ( $X, Y, Z$ ) in the LiDAR coordinate system. The LiDAR is installed on the side of the main road intersection on campus, where there are more non-motorized objects (such as pedestrians and bicycles) and fewer vehicle objects; therefore, it is suitable for pedestrian detection tasks. It should be noted that the installation position of the LiDAR affects the quality of the point cloud [30]. The installation position and height of the LiDAR are shown in Figure 9. The equipment was set on the street light pole on the side of the west entrance road with a good view. The street light pole is 20 m away from the intersection and

the radius is too small, only the isolated point describes the local feature, which reduces the classification effect and precision of Pointnet++. Hong et al. [28] proposed a Pointnet++ optimization algorithm fused with MKF. Based on the above algorithm, this study uses a variable-radius ball-query sampling mechanism. When there are no other points within the radius of the isolated point, the step size  $s$  is selected to increase the sampling radius of the ball query until the number of points in the radius reaches the threshold  $v$  ( $v > 1$ ). The sampling mechanism is illustrated in Figure 7.

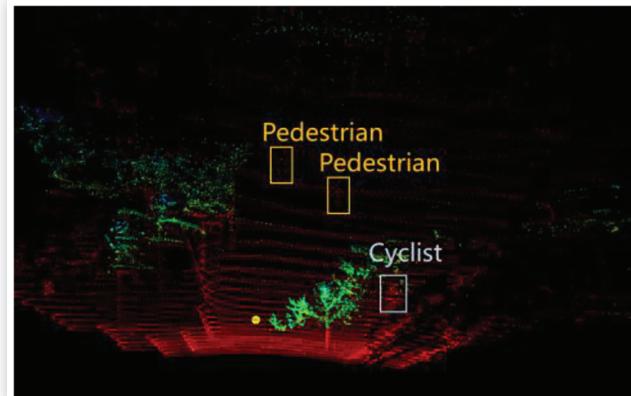
Figure 8 shows a comparison of the effect of the original sampling mechanism and the improved sampling mechanism on a sparse pedestrian point cloud. The improved sampling mechanism expands the sampling radius to obtain three points located on the legs of the pedestrian object as local area features, avoiding the use of isolated points to represent the features of the local area.

**FIGURE 8** Effects of improved SA layer.

© SAE International

**FIGURE 9** Installation of equipment.

© SAE International

**FIGURE 10** Point cloud collected by LiDAR.

© SAE International

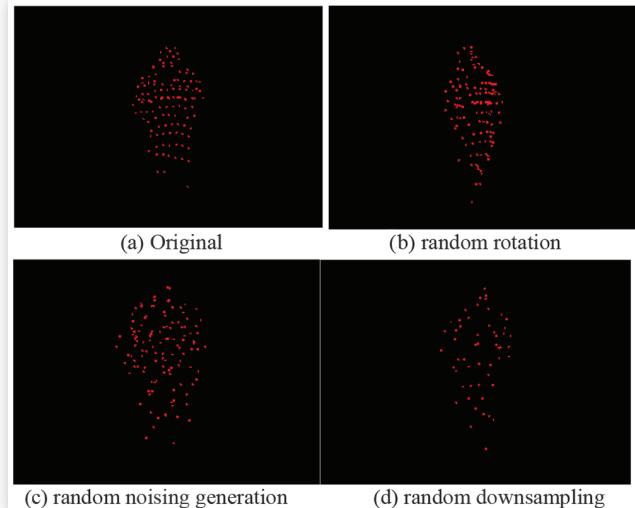
1.5 m from the curb. The installation height of the LiDAR was 6 ft, and the inclination angle was 15°.

## Dataset

This experiment used the equipment mentioned above to collect the data. To make the data more representative, the data was collected from different times on campus. In our study, different light conditions (sunny, cloudy) and different pedestrian densities (peak period and off-peak period) are covered. The data was collected for three days, from January 3, 2021, to January 5, 2021 (Saturday to Monday), and two hours of LiDAR data is collected from 11:00 to 13:00 every day. We screened representative 700 frames with pedestrians from the collected data in three days. The points in frames should keep the basic shapes and features of a pedestrian. Pedestrians, bicycles, and car objects in these 700 frames were manually labeled. These 700 frames included 3558 pedestrians, 157 vehicles, and 543 bicycles. In order to evaluate the robustness of our algorithm for the noises dataset, we manually added noising points to the collected data. To simulate noising points in real scenes, 50 noising points were added in every frame of LiDAR data and the distribution obeyed the literature [10].

## Data Augmentation

Owing to the limited amount of training data, training data is expanded using data enhancement to not affect the experimental effect. In the method proposed by Ghiasi et al. [31], training data underwent operations, such as duplication, random-scale jitter scaling, and random horizontal flipping, to improve the accuracy of image object detection. Based on the above method, this study replicated the training data to expand it up to three times and then performed random rotation, random noising generation, and random downsampling on the training data. As shown in Figure 11, the effect of data augmentation can help increase the amount of training data and improve the generalization ability and robustness of the model.

**FIGURE 11** Effects of data augmentation.

© SAE International

## Evaluation Index

The CPU used in the experiment is i7-6800K, with a memory size of 32GB, 3.4GHz of main frequency, NVIDIA GeForce GTX TITAN X graphics card, 12GB of video memory, and Pytorch 1.0.0 implementation tool. In the experiment, the AP was selected as the evaluation index of the pedestrian detection effect, and the calculation formula is as follows:

$$AP = \frac{1}{11} \sum_{r \subseteq \{0.0, 0.1, \dots, 1.0\}} \max(\tilde{r}) \quad \text{Eq. (5)}$$

where  $\tilde{r}$  is the recall of the detection algorithm.

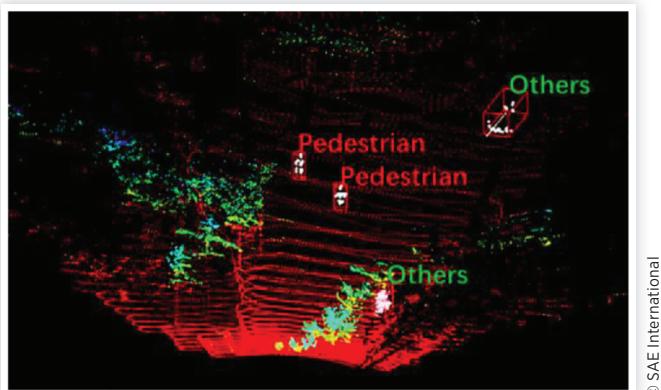
## Results

In the pedestrian detection effect experiment, we compared the effect of Pointnet++ before and after the improvement and the effect of comparison with other networks. We prepared original data and noising point clouds data for evaluation. All experiments were measured by AP. The results of the evaluation on original data are shown in Table 3. This method improves the AP by 5% compared with the original Pointnet++, which shows that the algorithm is effective in improving the detection precision of sparse point clouds. In addition,

**TABLE 3** Performance comparison in AP (%) on original data.

Method	Modality	AP
Voxelnet [17]	BEV	85.87
PointRCNN [20]	Point-wise	93.33
Pointnet++ [21]	Point-wise	89.88
Ours	Point-wise	<b>94.67</b>

© SAE International

**FIGURE 12** Detection results of our method.

compared with the results of Voxelnet and PointRCNN, our method is superior to the pedestrian detection results of these two networks. Figure 12 represents the result of pedestrian detection.

Table 4 shows the results of evaluation on noising points data. Though the performance of PointRCNN is close to our method on original data in Table 3, our algorithm shows robustness for noises data. With 50 noising points added to every non-ground object, our algorithm achieves an improvement of 5.58% compared with PointRCNN. The reasons for the result are as follows:

- PointRCNN is a two-stage method composed of RPN (Region Proposal Network) and refinement network. Obviously, noising points results in enormous difficulty to the part of foreground points segmentation and proposals generation.
- Our algorithm filtered out noising points from the background in the pre-filtering stages. For this reason, our algorithm is less influenced in training or inference in the noises dataset, and our algorithm is more robust compared with other methods [10].

In the above experiment, it was observed that selecting different step sizes  $s$  and threshold  $v$  affected the experimental results. The step size  $s$  determines the update of the sampling radius, and the threshold  $v$  determines whether to update the sampling radius.

Therefore, a parameter optimization experiment was designed to optimize the step size  $s$  and the threshold  $v$ . In the first part of the experiment, when the default step size  $s$  was

**TABLE 4** Performance comparison in AP (%) on noising points data.

Method	Modality	AP
Voxelnet [17]	BEV	80.61
PointRCNN [20]	Point-wise	84.73
Pointnet++ [21]	Point-wise	83.88
Ours	Point-wise	<b>90.31</b>

**TABLE 5** Results of parameter optimization experiment.

$v$	AP	$s(v=3)$	AP
Original	89.88	—	89.88
2	94.02	0.1	<b>94.67</b>
<b>3</b>	<b>94.67</b>	0.2	94.03
4	94.13	0.3	94.21
5	92.17	0.4	94.14

© SAE International

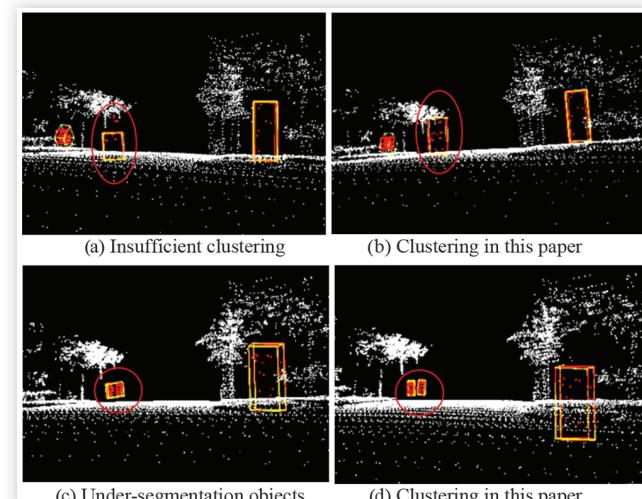
**TABLE 6** Performance comparison in runtime (ms).

Method	3D-DSF+Euclidean	Depth map+DBSCAN	Ours
Runtime (ms)	103.3	101.9	<b>88.7</b>

© SAE International

0.1, the size of the threshold  $v$  was increased from two to five, and the AP of the algorithm was compared to determine the optimal threshold  $v$ . In the second part of the experiment, the threshold  $v$  was set based on the final results of the first experiment, and step size  $s$  was increased from 0.1 to 0.5 to determine the optimal step size  $s$ . The results are shown in Table 5. When the threshold  $v = 3$  and step size  $s = 0.1$ , the algorithm performs best on the AP index.

In the clustering algorithm effect experiment, the real-time performance of the background filtering and clustering algorithms proposed in this study was verified. Table 6 compares the speed between the proposed clustering method and the current mainstream clustering method. The method in this study takes 88.7 ms per frame, which is approximately 1.16 times the time efficiency of 3D-DSF combined with Euclidean clustering algorithm, also it is approximately 1.14 times the time efficiency of DBSCAN improved by the depth map. In addition, the experiment tested the clustering effect on sparse point clouds and neighboring objects. Figure 13 shows a comparison between the original European clustering and the

**FIGURE 13** Comparison of the effects of clustering.

© SAE International

proposed clustering algorithm. After improvement, the problem of insufficient object clustering and under-segmentation of neighboring objects can be solved to a certain extent.

## Conclusion

In this study, a new pedestrian detection algorithm that combines traditional clustering methods and deep learning is proposed. Background filtering based on Octree increases the speed of the clustering process to 88.7 ms per frame. Based on an analysis of the relationship between the number of point clouds and their distance, this study proposes an adaptive distance threshold Euclidean clustering algorithm based on the characteristics of the LiDAR. The algorithm can adjust the search radius threshold based on the performance of LiDAR and distance between cloud points and LiDAR, which could improve the problem of insufficient clustering and under-segmentation of neighboring objects. In addition, the improvement of Pointnet++ in this study has been proven to be effective by experiment, and the AP of Pointnet++ on sparse point clouds is increased to 94.67%, which is better than the other two representative networks. In this study, the model obtained was deployed on NVIDIA Jetson AGX Xavier, attaining the inference time of 110 ms per frame, which met the requirements of real-time application and feedback of obstacle information during roadside LiDAR working. Meanwhile, our method has achieved high detection accuracy and practical application value. However, this method still requires further improvement for the under-segmentation of neighboring objects. When a crowded pedestrian queue appears at a distance, it exhibits a poor detection ability. We cut them out to ensure the result of the network in this study and try to classify them correctly by feature selection in applications. In most cases, this method works well but lacks reliability for engineering. In order to further improve the detection accuracy of a crowded pedestrian queue, we will attempt to use camera-LiDAR fusion methods to solve this problem. Further study will be conducted on how to construct a more complete pedestrian dataset, which lays a foundation for future research and development.

## Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant #SQ2020YFB160040 and the Beijing Natural Science Foundation under Grant #L191002. The authors would also like to thank the insightful and constructive comments from anonymous reviewers.

## References

- Express and Star, "Pedestrian Killed in Dudley Bus Crash Could Not Be Seen until It Was Too Late, Inquest Told," 2020, <https://www.expressandstar.com/news/local-hubs/dudley/2020/06/19/pedestrian-was-spotted-one-second-before-fatal-collision>, accessed 05 June, 2021.
- Sivaraman, S. and Trivedi, M.M., "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Transactions on Intelligent Transportation Systems* 14, no. 4 (2013): 1773-1795, <https://doi.org/10.1109/TITS.2013.2266661>.
- Ai, C. and Tsai, Y.J., "An Automated Sign Retroreflectivity Condition Evaluation Methodology Using Mobile LiDAR and Computer Vision," *Transportation Research Part C: Emerging Technologies* 63 (2016): 96-113, <https://doi.org/10.1016/j.trc.2015.12.002>.
- Du, X., Ang, M.H., and Rus, D., "Car Detection for Autonomous Vehicle: LiDAR and Vision Fusion Approach through Deep Learning Framework," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, IEEE, 2017, <https://doi.org/10.1109/IROS.2017.8202234>.
- Pranav, K.B. and Manikandan, J., "Design and Evaluation of a Real-Time Pedestrian Detection System for Autonomous Vehicles," in *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*, Novi Sad, Serbia, May, 2020.
- Li, C. et al., "Illumination-Aware Faster R-CNN for Robust Multispectral Pedestrian Detection," *Pattern Recognition* 85 (2019): 161-171.
- Zhao, J. et al., "Trajectory Tracking and Prediction of Pedestrian's Crossing Intention Using Roadside LiDAR," *IET Intelligent Transport Systems* 13, no. 5 (2018): 789-795.
- Fei, J. et al., "SemanticVoxels: Sequential Fusion for 3D Pedestrian Detection Using LiDAR Point Cloud and Semantic Segmentation," 2020.
- Tatebe, Y. et al., "Pedestrian Detection from a Sparse LiDAR Point-Cloud—Distribution-Based Voxel Representation for 3DCNN," *Journal of the Japan Society for Precision Engineering* 84 (2018): 1017-1024.
- Liu, Z. et al., "TANet: Robust 3D Object Detection from Point Clouds with Triple Attention," *Proceedings of the AAAI Conference on Artificial Intelligence* 34, no. 7 (2020): 11677-11684.
- Ibisch, A. et al., "Towards Autonomous Driving in a Parking Garage: Vehicle Localization and Tracking Using Environment-Embedded LiDAR Sensors," in *IEEE Intelligent Vehicles Symposium*, City of Gold Coast, Australia, IEEE, 2013.
- Ai, C. and Tsai, Y., "Automated Sidewalk Assessment Method for Americans with Disabilities Act Compliance Using Three-Dimensional Mobile LiDAR," *Transportation Research Record* 2542, no. 1 (2016): 25-32.
- Wu, J., Hao, X., and Zheng, J., "Automatic Background Filtering and Lane Identification with Roadside LiDAR Data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, IEEE, 2017.
- Ester, M., Kriegel, H.P., Sander, J. et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noising," Miinchen, Germany, *KDD (AAAI Press, 1996)*.
- Hartigan, J.A. and Wong, M.A., "A K-Means Clustering Algorithm," *Applied Statistics* 28, no. 1 (1979): 100-108.

16. Tian, Z., Ramakrishnan, R., and Livny, M., "BIRCH: An Efficient Data Clustering Method for Very Large," *ACM Sigmod Record* 25, no. 2 (1996): 103-114.
17. Yin, Z. and Tuzel, O., "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," 2017.
18. Lang, A.H. et al., "PointPillars: Fast Encoders for Object Detection from Point Clouds." 2018.
19. Qi, C.R. et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017.
20. Shi, S., Wang, X., and Li, H., "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, IEEE, 2019.
21. Qi, C.R. et al., "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," 2017.
22. Mao, J. et al., "Pedestrian Detection and Recognition Using LiDAR for Autonomous Driving," in *International Conference on Optical Instruments and Technology 2019: Optical Sensor and Applications*, Beijing, China, 2020.
23. Chang, S., Xu, Z., and Xuan, G., "Improvement of K Mean Clustering Algorithm Based on Density," 2018.
24. Li, X., Zhang, P., and Zhu, G., "DBSCAN Clustering Algorithms for Non-Uniform Density Data and Its Application in Urban Rail Passenger Aggregation Distribution," *Energies* 12 (2019): 3722.
25. Mosa, A.S.M. et al., "Evaluating the Benefits of Octree-Based Indexing for LiDAR Data," *Photogrammetric Engineering & Remote Sensing* 78, no. 9 (2012): 927-934.
26. Azim, A. and Aycard, O., "Detection, Classification and Tracking of Moving Objects in a 3D Environment," in *Intelligent Vehicles Symposium*, Alcala de Henares, Spain, IEEE, 2012.
27. Zhao, J. et al., "Detection and Tracking of Pedestrians and Vehicles Using Roadside LiDAR Sensors," *Transportation Research Part C: Emerging Technologies* 100 (2019): 68-87, <https://doi.org/10.1016/j.trc.2019.01.007>.
28. Hong, S. and Yue-lan, L., "Research on Pointnet ++ Optimization Algorithm Integrating MKF," *Journal of Chinese Computer Systems* 41, no. 6 (2020): 151-155.
29. Campos-Taberner, M., Romero-Soriano, A., Gatta, C., Camps-Valls, G. et al., "Processing of Extremely High-Resolution LiDAR and RGB Data: Outcome of the 2015 IEEE GRSS Data Fusion Contest—Part A: 2-D Contest," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing* 9, no. 12 (2017): 5547-5559, <https://doi.org/10.1109/JSTARS.2016.2569162>.
30. Zhao, J. et al., "Detection Range Analysis of Roadside LiDAR Sensors," in *Proceedings of the 98th Transportation Research Board Annual Meeting*, Washington, DC, January, 2019.
31. Ghiasi, G. et al., "Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation," 2020.