Traffic & Transportation

Global 3D Point Cloud Object Detection System Based on Data-Level Stitching

Yu LUO[1], Tao WANG[2], Shuai LU[3], Xuerui DAI[4], Zhi LI[5], Xuewei ZHANG[6]

Original Scientific Paper
Submitted: 25 July 2024
Accepted: 20 Jan 2025

[1] 22656661@qq.com, Sichuan Expressway Construction Development Group Co., Ltd., Sichuan, China
[2] wangtao@wanji.net.cn, Department of Intelligent Internet Connection, Beijing VANJEE Technology Co., Ltd., Beijing, China
[3] lushuai@wanji.net.cn, Department of Intelligent Internet Connection, Beijing VANJEE Technology Co., Ltd., Beijing, China
[4] Corresponding author, daixuerui@wanji.net.cn, Department of Intelligent Internet Connection, Beijing VANJEE Technology Co., Ltd., Beijing, China
[5] lizhi@wanji.net.cn, Department of Intelligent Internet Connection, Beijing VANJEE Technology Co., Ltd., Beijing, China
[6] zhangxuewei@wanji.net.cn, Department of Intelligent Internet Connection, Beijing VANJEE Technology Co., Ltd., Beijing, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
With the rapid development of artificial intelligence, the application prospects of global perception systems that can cover large-scale smart tunnel scenes are becoming increasingly widespread. Using multi-modal data from different sensors, the global perception system attempts to locate and track traffic targets. Due to the presence of detection blind spots at a considerable distance between two stations, which increases the difficulty of detection, the conventional global stitching method based on result-level stitching easily leads to problems such as lost vehicles and discontinuous trajectories in the blind area, and the low detection accuracy of sparse point cloud detection at the single station. To address these issues, this paper optimised the point cloud detection algorithm by improving the network structure and loss function to enhance the perception capability of the single station. Additionally, it proposed a data-level global point cloud stitching algorithm and a method for sampling from a difficult database, replacing the traditional global result-level stitching method and ensuring the fusion effect of global trajectories. Overall, this provides a more reliable and comprehensive perception fusion result for platform twinning. Finally, to validate the effectiveness of our method, we introduced the publicly available VANJEE-PointCloud dataset collected in the real world. The experiments show that our algorithm not only enhances perception capability but also improves the success rate of global trajectory fusion.

KEYWORDS
3D object detection; sparse convolution; atrous convolution; feature aggregation.

1. INTRODUCTION

In recent years, with the advancement of deep learning and sensor technology, significant progress has been made in roadside global perception algorithms [1, 2], in which point cloud-based global perception is a vital technology. The point cloud-based global perception task takes multi-lidar data as input and outputs locations and trajectories of objects. To meet the global perception needs in various scenarios, global perception systems have been proposed. These systems typically consist of multiple independent intelligent base stations, which transmit their respective perception results to the global perception system through networks. The global perception system can intelligently process and analyse perception data, automatically identify and classify targets, and extract valuable information to meet practical requirements. However, there are challenges in data

connection and sharing between multiple base stations. Due to factors such as time synchronisation between multiple base stations and high-precision positioning of detection frames, existing algorithms suffer from matching deviations and discontinuities in trajectories at the intersections of base stations. This paper mainly focuses on issues related to global perception systems in multi-base station deployment scenarios based on lidar point clouds.

In global perception systems, commonly used sensors include cameras, millimetre-wave radars and lidars. Lidars have the advantages of high precision and high resolution, which can accurately locate the position and height information of target objects with a precision of up to centimetres. Therefore, they are widely used in the field of target perception. This paper focuses on global perception systems based on lidars. Point cloud target perception algorithms have evolved from early geometric feature-based clustering algorithms to widely researched deep learning-based object detection algorithms. However, the scanning principle of mechanical lidars leads to sparse point clouds in the distance, and targets often occlude each other, making it difficult to perform effective detection.

To address the aforementioned challenges, we have proposed a global perception solution based on a point cloud object detection algorithm. Firstly, to address the data connection issues between multiple base stations, we have employed a data-level optimisation strategy, namely data-level point cloud stitching (DLS). This strategy is also applicable to other sensor-based object detection tasks, and during training, we have introduced a "hard database" sampling data augmentation method. Secondly, in order to extract richer, deeper features and effectively combine these features, we have designed a deep and attention feature aggregation (DAFA) module for two-dimensional feature extraction. Finally, to better regress the size and orientation of target boxes, we have introduced a new loss function called comprehensive distance-IOU(CDIoU) loss. We have conducted comparative experiments with other similar algorithms under the same experimental settings on a public dataset that we are about to release. Our point cloud perception algorithm has shown improvements in recall rate, recognition accuracy and global trajectory perception rate. Additionally, we have conducted ablation experiments to demonstrate the effectiveness of each optimisation module. The contributions of our work can be summarised as follows:

1) Data-level: We have enriched the point cloud features of distant targets and improved the fusion rate of global trajectories. We have proposed a data-level global point cloud stitching algorithm and "hard target database" sampling.
2) Network structure: In order to extract richer, deeper two-dimensional features, we have proposed the DAFA module.
3) Loss function: For better regression of target size and orientation, we have introduced the CDIoU loss.

2. RELATED WORKS

Below, we briefly review existing works on 3D object detection based on global perception algorithms and point cloud perception algorithms. These works motivate us to focus on developing a global perception method based on point cloud.

2.1 Global perception algorithm

The global perception algorithm is based on artificial intelligence and communication technology. Usually, a set of global perception systems can be composed of multiple intelligent base stations. Through the collaboration and information sharing between base stations, real-time perception and monitoring algorithms for all traffic participants within the coverage area of the base station are realised. Its accuracy mainly depends on the output of its upstream single base station. Firstly, the global algorithm requires the detection result data of different base stations to be mosaic. This step requires high-precision positioning of the base station to avoid stitch errors. At the same time, this stitching method also requires a good network environment to achieve real-time acquisition of detection results from all base stations. Although the application of multi-base station and global perception algorithms can increase the perception range, the detection result-level stitch will ignore the long-tail distribution and data complementarity problems of lidar. Finally, from the perspective of algorithm

complexity, data-level stitching only requires joint data, while the result-level mosaic method needs to pay attention to the position, heading angle and category, and its complexity is significantly higher than that of data-level stitching.

2.2 Point cloud perception algorithm

Unlike image data, due to the hardware features of lidar, its data have strong sparsity and disorder. For object detection based on point clouds, the advantage is that it can use three-dimensional spatial information, but the disadvantage is that the increase in spatial dimensions often leads to point cloud data that is too sparse, resulting in poor model fitting [3].

An early common approach is to convert 3D point cloud data into 2D image form for input. Using this approach, researchers can directly apply traditional 2D object detection algorithms to 3D point cloud data detection tasks. For example, the MVF (multi-view fusion) [4] algorithm is an effective end-to-end multi-view fusion (MVF) algorithm that converts 3D point cloud data into multiple views through dynamic voxelisation and utilises the features of multiple views for object detection. The front view representation in the deep learning direction includes projections such as depth images and spherical projections. However, this method loses the 3D features of the point cloud.

Later, algorithms directly based on 3D point clouds were proposed. PointNet [5] is a neural network-based end-to-end point cloud classification and segmentation method proposed by Charles R. Qi et al. in 2017. PointNet directly takes point clouds as input and captures local and global feature information in point cloud data through high-dimensional mapping and maximum pooling operations. This method can better express the 3D information of point clouds. However, PointNet has a major problem: it can accommodate a small number of point clouds, so it is more often applied in semantic segmentation or object detection of small targets and is not suitable for large-scale point cloud applications such as intelligent transportation. Bird's eye view (BEV) represents point cloud data better than deep maps. It represents point clouds from a top-down perspective without losing any scale and scope information and is widely used in lidar detection [6-10], which has also been recently used for task segmentation [11]. PointPillars [12] is improved by adding a PointNet model to the BEV representation. PointNet is used to convert the point cloud in each column grid into a fixed-length vector, forming a pseudo image. Then, a 2D convolutional neural network is used for feature extraction and object detection operations. PIXOR [13] discretises the point cloud into a BEV representation and encodes the features of each cell as occupancy and normalised reflectance. Next, a neural network with 2D convolutional layers is used for 3D object detection.

In addition to directly processing 3D point cloud data, some researchers have also converted 3D point cloud data into voxel form for input. VoxelNet [14] is a network structure that converts 3D point cloud data into voxel form for input. It uses 3D convolution to process voxelised point cloud data to achieve object detection. A similar idea is also adopted, such as Frustum PointNet [15] and PointNetVLAD [16]. SECOND algorithm is a target detection algorithm based on 3D point cloud voxels, with the full name of sparsely embedded convolutional detection. The design idea of this algorithm is almost identical to that of VoxelNet, with the main difference being that the convolutional middle layer (CML) in VoxelNet is replaced by 3D sparse convolution for feature extraction. By using submanifold convolution, the "inflation" problem that occurs when processing data in dense convolution is solved. Based on neural network-based point cloud perception algorithms, this article mainly focuses on optimising SECOND as a benchmark.

In general, although point cloud object detection algorithms have been widely applied in many fields, there are still many challenging problems that need to be addressed. For example, how to effectively process large-scale 3D point cloud data and accurately detect various shapes and sizes of target objects. We will study and optimise these problems in this paper.

3. APPROACH

In this section, we describe the overall structure of the proposed global perception algorithm. We further introduce the main innovations in detail, including "point cloud stitching", "hard database", "DAFA module" and "CDIoU loss".

3.1 Overall framework

The innovation of this work mainly lies in three aspects: data stitching and database sampling, the DAFA module and loss function. The network structure is as follows. Firstly, the high-precision position information of base stations is utilised to stitch the data from multiple base stations, forming a point cloud dataset covering the detection ranges of multiple base stations, with data augmentation through "hard database" sampling during training. Next, the point cloud data are fed into a point cloud voxel feature extraction network, which encodes the discrete and unordered 3D point cloud into a sparse 4D tensor. Sparse convolution is used in the 3D feature extraction module. Finally, we propose the DAFA module for two-dimensional feature extraction. For target box regression, we introduce a new loss function called comprehensive distance-IOU, which mainly includes IoU loss, distance loss, aspect ratio loss and angle loss. The overall network structure is shown in *Figure 1*. In the following sections, we will provide detailed explanations of the work we have done in model optimisation.

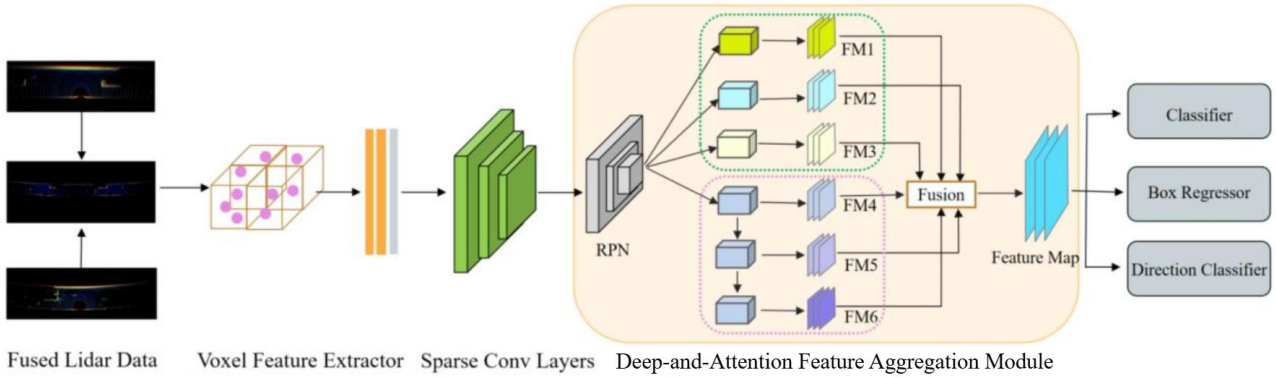


Figure 1 – The structure of our proposed algorithm. The detector takes a raw point cloud as input, converts it to voxel features, and applies two VFE (voxel feature encoding) layers following a linear layer. Then, a sparse CNN is applied. Finally, the DAFA module and detection head generate the detection.

3.2 Data-level point cloud stitching and sample ground truth from the "hard database"

The traditional global stitching methods are based on the result-level backend stitching where perception results of multi base stations are sent to the global perception system to be fused and stitched, which is limited with the performance of single base station. In contrast, we present the data-level point cloud stitching strategy, with which the network is able to directly consume fused point clouds and sufficiently dig out the complementarity of multi-lidar data. This strategy contains two procedures, multi-lidar pose calibration and multi-lidar data stitching.

Pose calibration. Let $\{L_i: i = 1, \dots, N\}$ be the lidar set in a tunnel scene, N is the number of lidars, L_i represents the i-th lidar. Timestamps of point clouds from these lidars have been synchronised. While L_1 is chosen as the primary lidar, the others are regarded as deputy lidars. The relative pose between each deputy lidar and primary lidar needs to be calculated so that all of the point clouds can be projected to the primary lidar's coordinate correctly.

For a pair of adjacent lidars L_k and L_{k+1}, the points in their overlapped scanning area are used for calibration. As illustrated in *Figure 2*, in our "LidarCalibration" software, we could manually adjust the translation parameters [translation x, translation y, translation z] and the rotation parameters [roll, pitch, yaw] of the L_k lidar's coordinate. In the overlapped area, each object's shape is always described by two-point sets in different directions. When the two point sets of each reference object are aligned correctly, the calibration between the two lidars is completed. At the same time, the pose transformation $T_{L_k L_{k+1}} \in R^{4\times4}$ which indicates how L_k coordinate moves to L_{k+1} coordinate, is naturally calculated base on the translation parameters and rotation parameters. Let $\{T_{L_k L_{k+1}}: k = 1, \dots, N-1\}$ be the pose transformation set, we further transform it to $\{T_{L_1 L_{k+1}}: k = 1, \dots, N-1\}$ which is applied for converting point clouds to primary coordinate (e.g. $T_{L_1 L_p} = T_{L_1 L_2} \cdot T_{L_2 L_3} \cdot \dots \cdot T_{L_{p-1} L_p}$ represents the transformation from the primary lidar L_1 to the deputy lidar L_p).

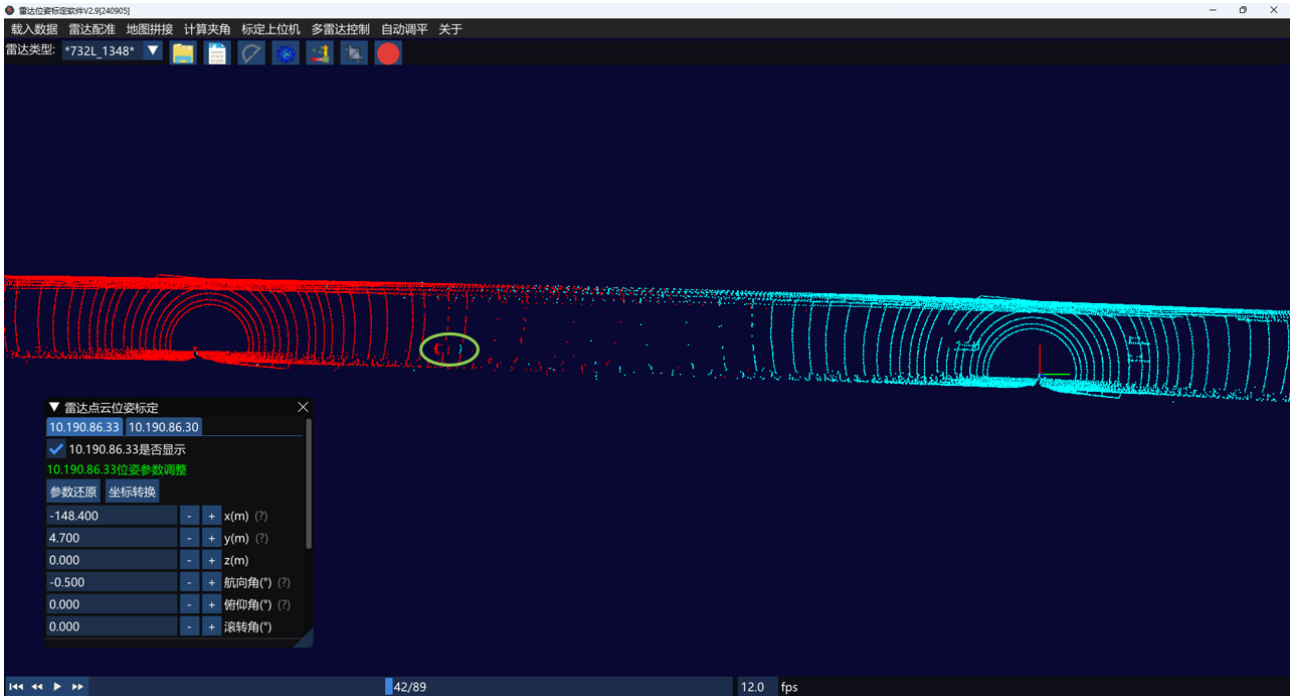


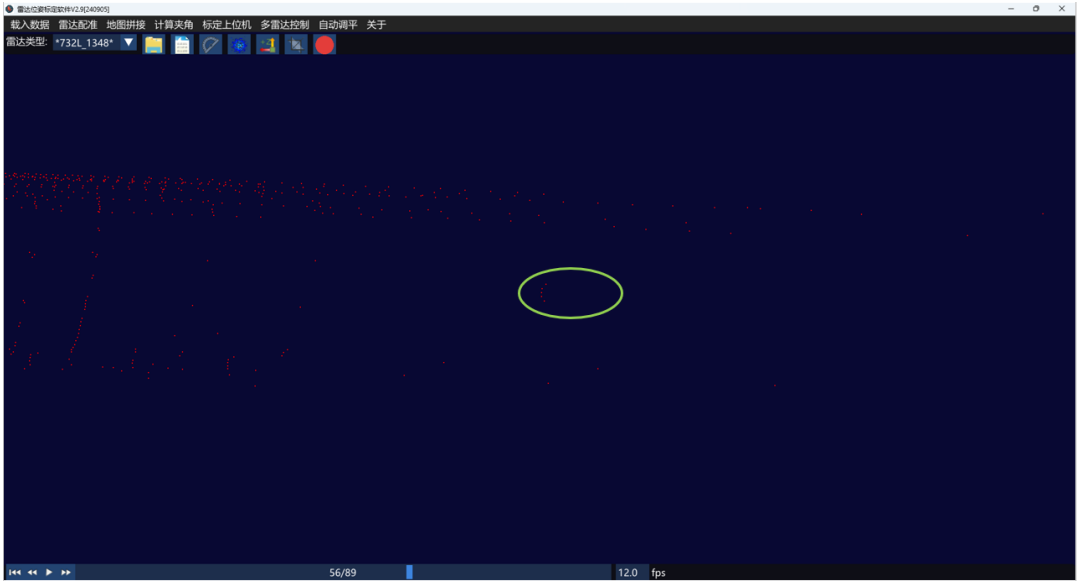
Figure 2 – Manual pose calibration

Point cloud stitching. A point cloud captured by lidar L_k in initial coordinate system is noted as $P_{L_k}^k = \{p_i = (x_i, y_i, z_i, intensity_i): i = 1, \dots, M\}$, M is the number of points. It will be converted to $P_{L_1}^k$ which in the primary lidar L_1 coordinate system by:

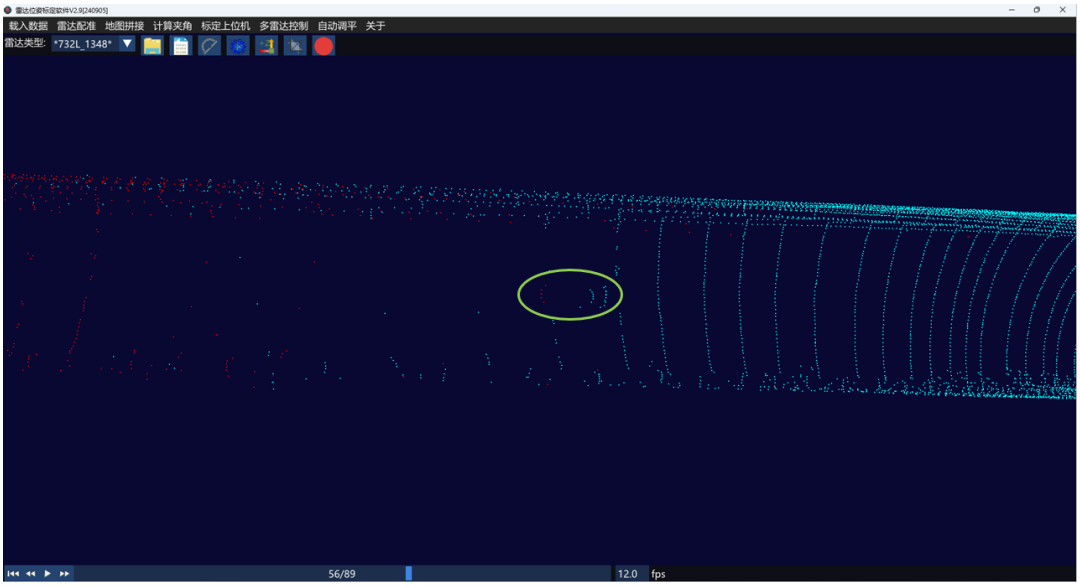
$$P_{L_1}^k = T_{L_1 L_k} \times P_{L_k}^k \tag{1}$$

where × means matrix multiplication. During calculation, $intensity_i$ is replaced with the value 1 for a homogeneous coordinate representation. After that, $intensity_i$ is remapped into p_i.

We concatenate all of the transformed point clouds directly to obtain a global point cloud, which is fed to the object detection network to predict 3D bounding boxes. Benefited from the complementarity of multi-lidar data, the shapes of objects in a global point cloud are more detailed than those in a single point cloud, which significantly improves the detection accuracy. Especially for blind detection areas. In a single lidar's point cloud, a blind detection area means a region where observed points are rare and objects are unable to be detected successfully. Specifically, as illustrated in *Figure 3*, point cloud stitching effectively increases the number of points in the blind detection area and complements the shapes of objects so that the network can do detection more easily.



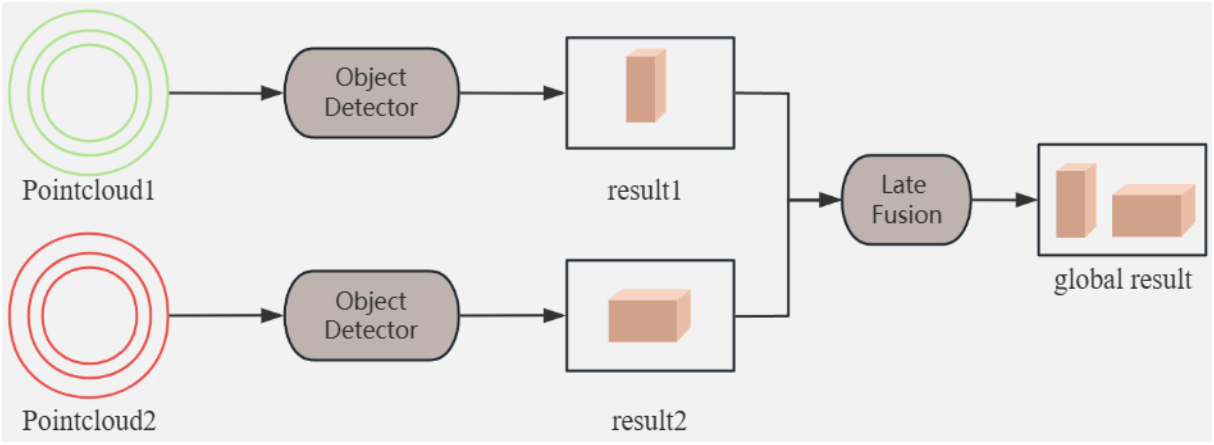
(a) A blind area in a point cloud



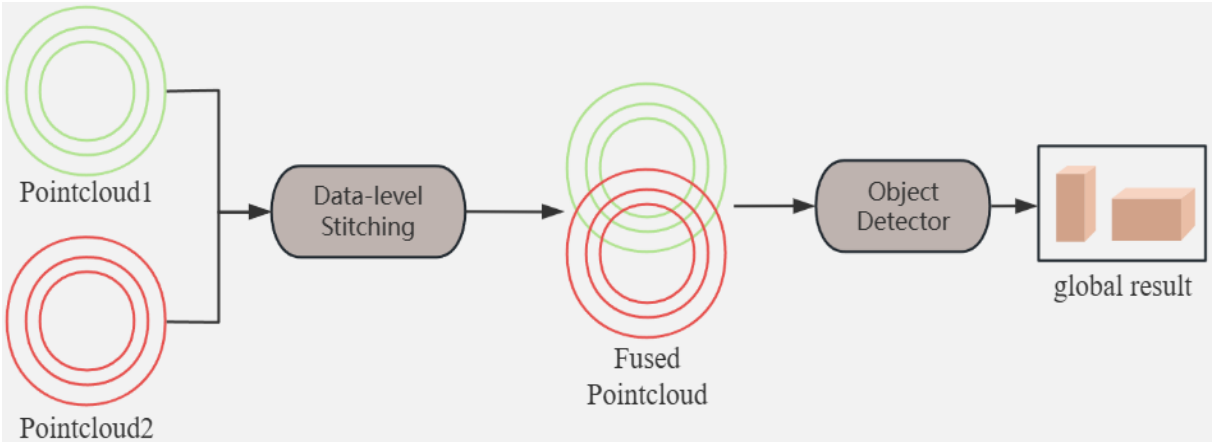
(b) The same area in a stitching point cloud

Figure 3 – Data-level stitching increases the number of points in the blind area

At the same time, using the method of multi-lidar data stitching, the point cloud detection algorithm can process multiple lidar data simultaneously, sending multiple lidar detection results to the global perception system at once. Therefore, the number of perception result stitches in the global perception system is reduced, and the reliability of the trajectory is increased. *Figure 4* shows the traditional data processing method and the new processing method proposed in this article.



(a) The traditional global perception system



(b) The proposed global perception system based on data-level stitching

Figure 4 – The above image depicts the traditional global perception system, while the below image illustrates the new global perception system proposed in this article

In addition, we establish a "hard database" for data augmentation during the training. Firstly, the detection network was trained on the training set of VANJEE PointCloud, there is an existing "regular database", as mentioned in SECOND [8]. Then, we let the trained network do inferences on the training data, and we generate the "hard database" containing the labels of all undetected targets and their associated point cloud data. We further train the trained network on the training set again for fine-tuning. During the fine-tuning, "regular database" is replaced with "hard database", and several ground truths from "hard database" are sampled to be introduced into the current training point cloud via concatenation. Using this method, the network could counter more hard-level objects and learn to recognise them.

3.3 Deep-and-attention feature aggregation module

One of the important innovations in this paper is the proposal of the DAFA module. The DAFA module helps extract more robust features with rich spatial and semantic information for more accurate predictions of bounding boxes and classification confidence, as shown in *Figure 5*.

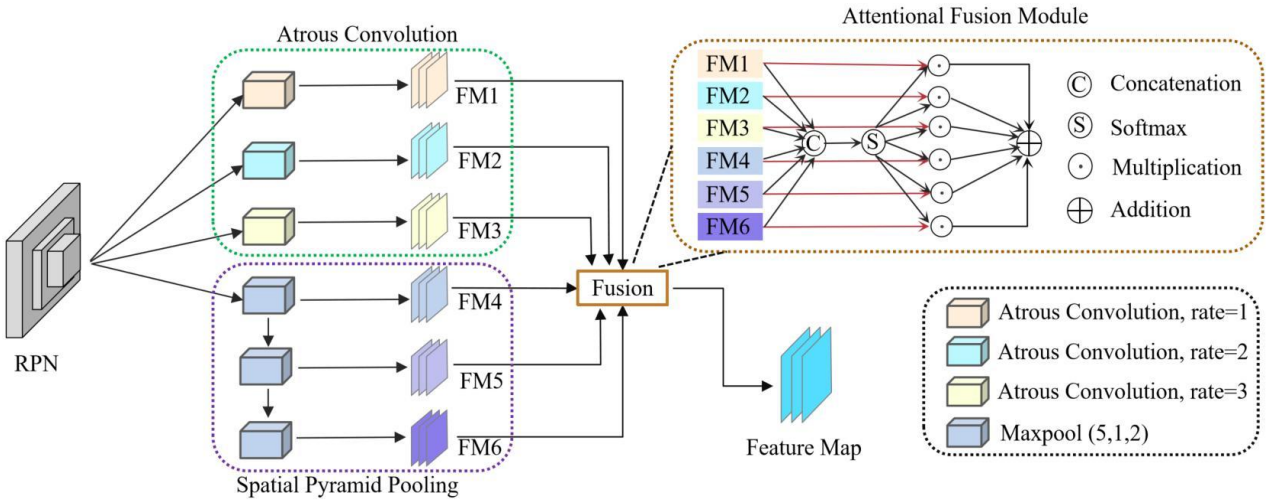


Figure 5 – Overview of the proposed DAFA module, which consumes the feature map outputted by the region proposal network (RPN). First, we use dilated convolutions with different dilation rates and spatial pyramid pooling for further feature extraction. Then, we fuse the features to generate the final feature map, incorporating an attention mechanism during the fusion process.

To achieve a larger receptive field without significantly sacrificing resolution, one can utilise larger convolutional kernels or employ larger strides in pooling operations. However, the former approach leads to increased computational complexity, while the latter results in resolution loss. Balancing the desire for a larger receptive field in feature extraction while maintaining relatively high resolution poses a challenge due to this inherent trade-off. Dilated convolutions, also known as atrous convolutions, offer a solution to this dilemma. They allow for an expanded receptive field without substantial resolution loss.

In this context, employing three successive dilated convolutions with 3x3 kernels and dilation rates of 1, 2 and 3, respectively, enables the extraction of multi-scale information. The resulting receptive fields are 3, 5 and 7, respectively. This approach facilitates the preservation of feature map dimensions while leveraging multi-scale information, thereby avoiding the information loss associated with down-sampling operations. The calculation of the receptive field is as follows:

$$F = (k-1) \cdot r + 1 \tag{2}$$

where F denotes the receptive field, k represents the kernel size, and r stands for the dilation rate.

Besides the dilated convolutions, DAFA utilises spatial pyramid pooling with different receptive field sizes applied to the input feature map to obtain multi-scale feature representations. These feature representations capture object information at different scales, thus improving the model's detection performance for objects of various sizes. By integrating information from multiple scales, this module further enhances detection performance and robustness.

To adaptively fuse the enriched spatial feature and the upsampled semantic feature, we adopt the attentional fusion module. Let $\{F_i \in R^{H \times W \times C} : i = 1, \ldots, 6\}$ represent the six feature maps from the atrous convolution module and spatial pyramid pooling module, where C is the number of channels. First, we compress the channels of each feature map to one and concatenate them along the channel axis as $F_i \in R^{H \times W \times 6}$. Then we use $Softmax$ function to normalise the six concatenated channels and split them into six weight maps as follows:

$$F_{softmax} = Softmax(F_{cat}) \in R^{H \times W \times 6} \tag{3}$$

$$W_i = \lfloor F_{softmax} \rfloor_i \tag{4}$$

where the operator $\lfloor \cdot \rfloor_i$ means extracting the i-th dimension data along the C axis. $Softmax$ function builds the dependence between the six features for adaptive feature fusion. The final fused feature map F_{fuse} can be calculated by:

$$F_{fuse} = (F_1 \cdot W_1) \oplus \ldots \oplus (F_6 \cdot W_6) \tag{5}$$

where $\cdot$ is element-wise product with broadcast and $\oplus$ is element-wise addition. The fused feature F_{fuse} is finally sent to a detection head.

3.4 Comprehensive distance-IOU

When using labels for supervised training, the smooth-L1 loss is commonly used to constrain the regression of target boxes. However, due to long distances and occlusions in outdoor scenes, it is difficult to obtain sufficient information from sparse points to accurately predict the size of target boxes. In order to more accurately predict the position and orientation of target boxes, researchers have designed the orientation-aware distance-IoU loss (ODIoU) [17], which focuses more on the alignment of the centre point and orientation between the predicted box and the ground truth. The formula of the loss function is shown in *Equation 6*

$$L_{box} = 1 - IoU\left(B_p, B_g\right) + \frac{c^2}{d^2} + \gamma(1 - |\cos(\Delta\theta)|) \tag{6}$$

where B_p and B_g denote the predicted and ground-truth bounding boxes, respectively, c denotes the distance between the 3D centres of the two bounding boxes (see in *Figure 6*), d denotes the diagonal length $|AC|$ of the minimum cuboid that encloses both bounding boxes; $\Delta\theta$ denotes the BEV orientation difference between B_p and B_g; and γ is a hyper-parameter weight.

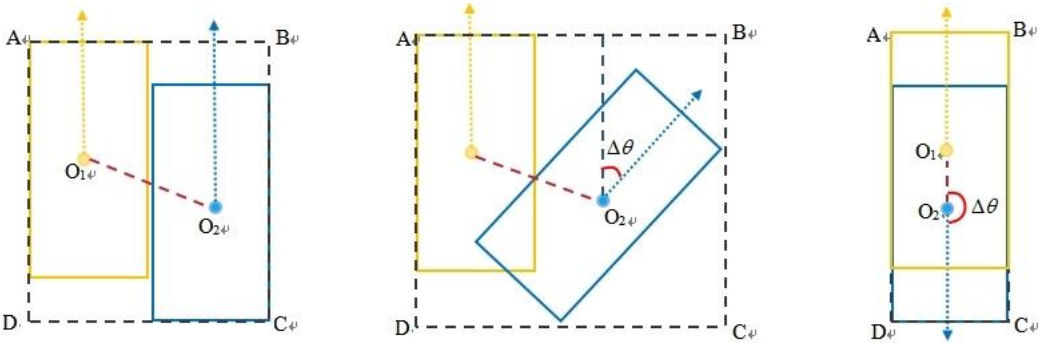


Figure 6 – Illustration of the disparity between predicted bounding boxes and ground truth in terms of intersection over union (IoU). Yellow boxes represent ground truth, blue boxes represent predicted bounding boxes. $|O_1O_2|$ denotes the distance between centre points, $\Delta\theta$ is the orientation difference in BEV.

However, it is noted that the ODIoU loss function still has drawbacks. Firstly, this function does not take into account the differences in length and width between the predicted box and the ground truth. As shown in *Figure 7*, when the predicted box has different length and width but the same area and angle, the loss function values are the same. Secondly, this loss function cannot distinguish the angle differences between 0 and 180. As shown in *Figure 8*, when the angle is 0 or 180, the angle loss value is always 0.

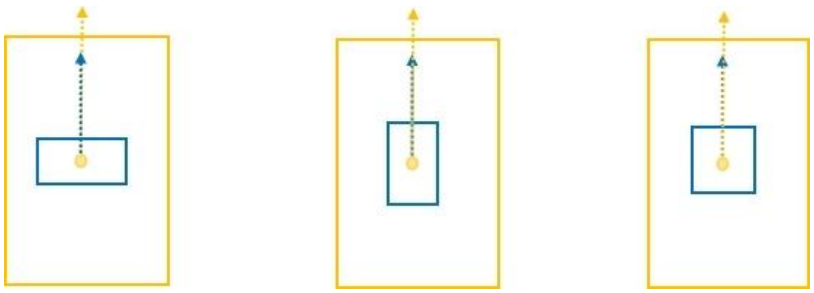


Figure 7 – Illustration of targets with different aspect ratios (yellow bounding boxes represent ground truth, blue bounding boxes represent predicted results

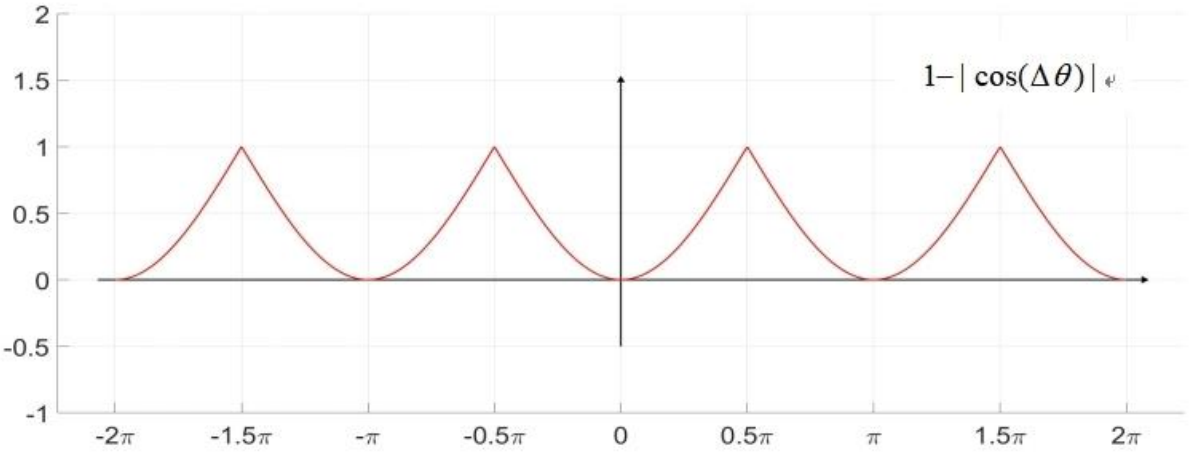


Figure 8 – Graph of the $(1 - |cos\ (\Delta\theta)|)$*, $\Delta\theta$ denotes the BEV orientation difference between B_p and B_g, when the angle is 0 or 180, the angle loss value is always 0*

To address the above two issues, we propose the comprehensive distance-IOU loss (CDIoU). This loss function retains the original IoU loss and distance loss of ODIoU, adds length and width losses, and optimises the angle loss. The formula is shown in *Equation 7* as follows:

$$L_{box} = 1 - IoU(B_p, B_g) + \frac{c^2}{d^2} + \frac{(l_p - l_g)^2}{l^2} + \frac{(w_p - w_g)^2}{w^2} + \gamma(1 - |\cos(\frac{\Delta\theta}{2})|) \tag{7}$$

where the difference from ODIoU is that our loss values gradually increase within the range of 0 to 180, as shown in *Figure 9*, allowing for better differentiation in cases involving 0 and 180 angles. B_p and B_g denote the predicted and ground-truth bounding boxes, respectively, c denotes the distance between the 3D centres of the two bounding boxes (see in *Figure 6*), d denotes the diagonal length $|AC|$ of the minimum cuboid that encloses both bounding boxes; l_p denotes the length of the predicted bounding box; l_g denotes the length of the ground-truth bounding box; l denotes the length $|BC|$ of the minimum cuboid that encloses both bounding boxes; w_p denotes the width of the predicted bounding box; w_g denotes the width of the ground-truth bounding box; w denotes the width $|AB|$ of the minimum cuboid that encloses both bounding boxes; $\Delta\theta$ denotes the BEV orientation difference between B_p and B_g; and γ is a hyper-parameter weight.

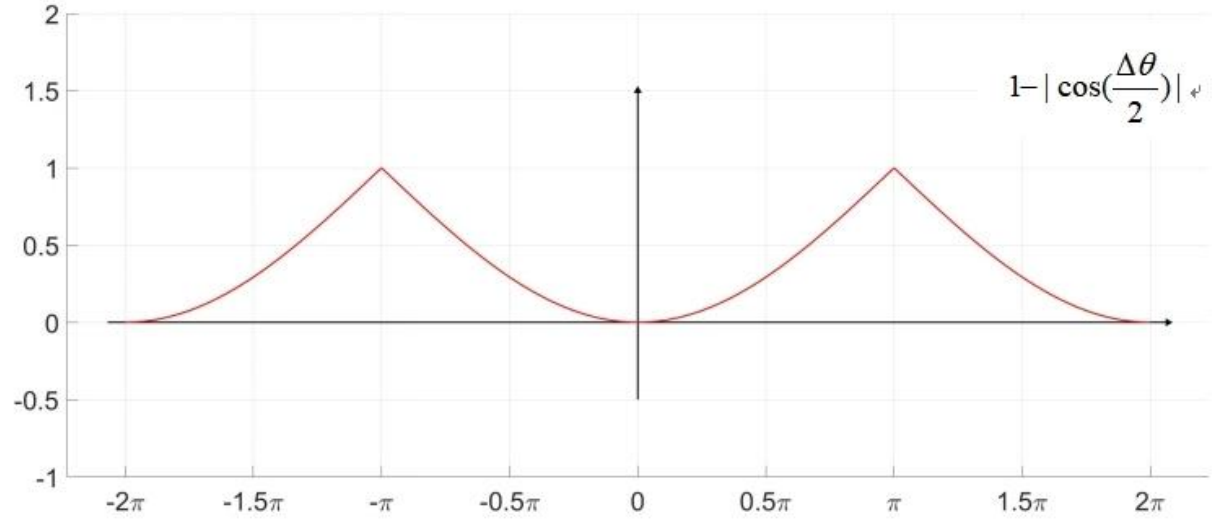


Figure 9 – Graph of the $(1 - |cos\ (\frac{\Delta\theta}{2})|)$*, $\Delta\theta$ denotes the BEV orientation difference between B_p and B_g, the gradient gradually increases as the angle ranges from 0 to 180*

Besides, we use the Focal loss and cross-entropy loss for the bounding box classification (L_{cls}) and direction classification (L_{dir}), respectively. The classification task is optimised by Focal loss, i.e

$$L_{cls} = -\alpha(1-s)^{\gamma}\log(s) \tag{8}$$

where

$$s = \begin{cases} s_p & if\ s_g = 1 \\ 1 - s_p & otherwise \end{cases} \tag{9}$$

s_g is a binary label to indicate whether an anchor box is a positive sample. s_p is the positive probability predicted by the network. α and γ are the hyper-parameters and are set to 0.25 and 2, respectively. For the direction classification task, we use the following method to generate the direction classifier target: if the yaw rotation around the z-axis of the ground truth is higher than zero, the result is positive; otherwise, it is negative. Hence, the overall loss to train is:

$$L = \beta_1 * L_{cls} + \beta_2 * L_{box} + \beta_3 * L_{dir} \tag{10}$$

where $\beta_1 = 1.0$, $\beta_2 = 2.0$, $\beta_3 = 0.2$ are constant coefficients of the loss formula.

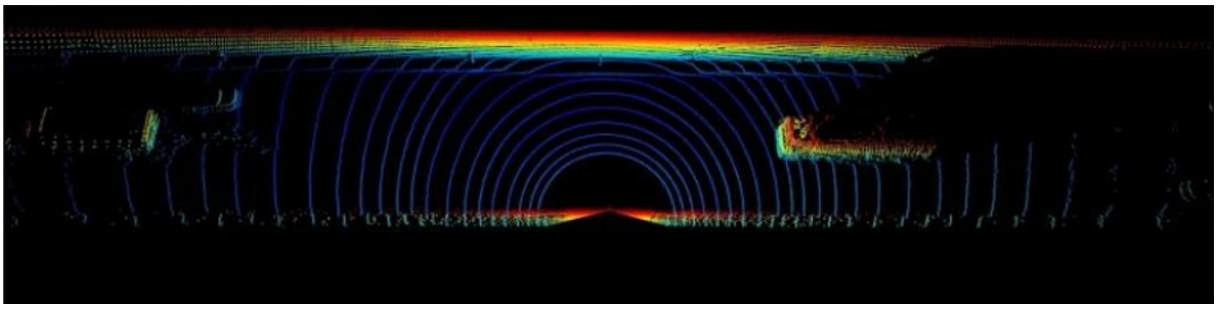
4. EXPERIMENTS

In this section, we use the upcoming public dataset to validate our innovative method. We will describe the experimental background and experimental methods in detail. We further demonstrate the effectiveness of our innovative points through ablation experiments and compare them with the latest algorithm models.

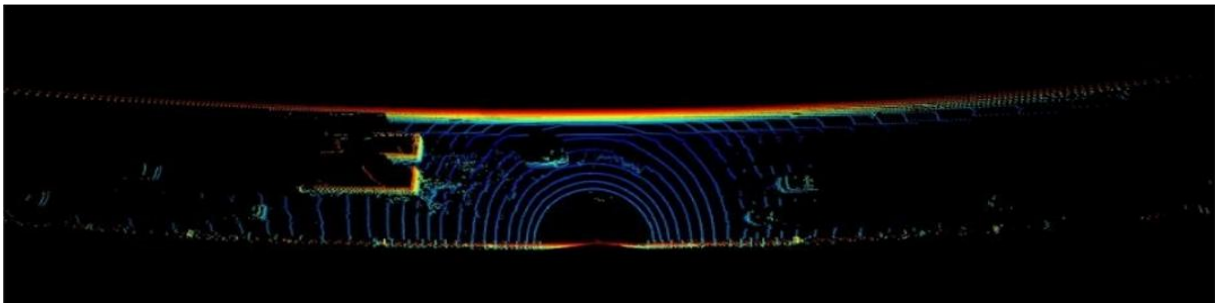
4.1 Dataset

In order to obtain a reasonable and effective method for evaluating results, we collected point cloud data from real tunnel environments as VANJEE PointCloud, which contains 12 unique scenes. VANJEE PointCloud includes 12 exclusive scenes, where the point cloud data have been carefully annotated in 6 classes, including car, bicycle, bus, tricycle, person and truck. *Figure 10* visually presents the point cloud data sample from two stations. Point clouds were captured by multi 32-line lidars, points from different lines have different colours.

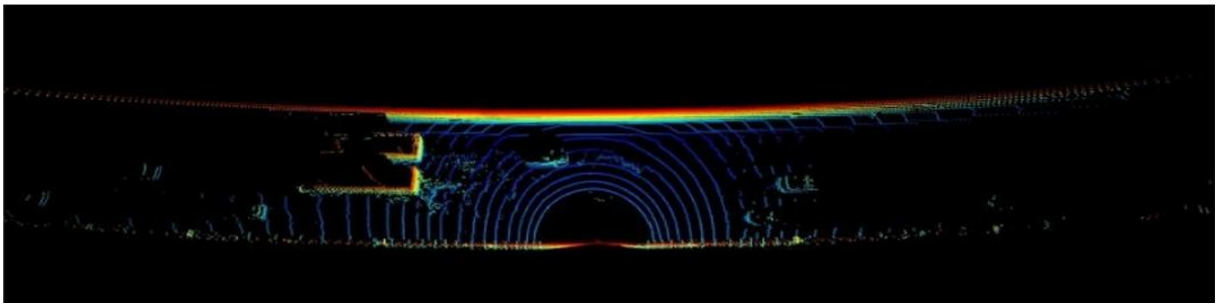
For point cloud detection tasks, we use the VANJEE PointCloud dataset to evaluate our method. Regarding all of the collected point clouds in a tunnel at the same time as a unit, we randomly split the VANJEE PointCloud into the training set and testing set. The training set includes 30,000-point clouds and the testing set includes 4000-point clouds. VANJEE PointCloud contains 6 classes, with varying numbers of object counts per class ranging from 2,000 to 110,000. This data distribution reflects the long-tail effect observed in real-world scenarios. We use a portable point cloud annotation platform system [18] to obtain ground truths. The dataset annotation is based on the Kitti format. Each line in the annotation file represents the label information of the object, including the category of the object, the occlusion level, the direction angle, the 3D centroid coordinates and the dimensions of the 2D and 3D bounding boxes. For detailed reference, please refer to the Kitti dataset format [19].



(a) The point cloud of station A



(b) The point cloud of station B



(c) The stitched point cloud of station A and station B

Figure 10 – The point cloud data sample from two stations. Figure (a) and figure (b) represent point clouds from two stations. Figure (c) shows the stitching result.

4.2 Experimental setting

During model training, we adopted a batch training method with a batch size of 8. The initial learning rate was set to 0.005, and an exponential decay learning strategy with a decay factor (gamma) of 0.95 was adopted. The training period was set to 200 epochs. We use the matching IoU thresholds for the positive and negative anchors of 0.6 and 0.45, respectively. The matching IoU between the bounding boxes and anchors is calculated by their nearest horizontal rectangles in BEV. Considering the actual size of the object, different anchor sizes were assigned to each object class, excluding all anchors corresponding to empty voxel points. In the testing stage, a threshold of 0.2 was used to filter out prediction boxes with low confidence scores, and non-maximum suppression (NMS) with a threshold of 0.3 was applied. In order to alleviate the detection performance problem caused by long-tail effects in the data, we additionally sample several ground truths of rare categories and apply random rotation and translation operations to them, then we add them into the “regular database”.

Due to the fact that the point cloud algorithm research in this article is based on mechanical lidar, it can be inferred from the characteristics of lidar that an object has detailed differences in the direction of lidar arrival and departure. Therefore, the original features of the object at the single radar and where multiple radars intersect are different. However, we can address this issue by concatenating and enhancing the training data as well. In terms of data augmentation, this article chooses strategies such as randomly increasing the number of samples at the splicing point within the lidar detection range and cropping the angle of the target part.

4.3 Comparison with state-of-the-art

In order to validate the effectiveness of the proposed object detection model in this paper, we conducted comparative experiments between the current mainstream lidar point cloud models and the model proposed in this paper. The dataset used in the experiments is the public dataset mentioned in Section 4.1. We use public implementations of those existing state-of-the-art methods, mostly available in a 3D object detection framework named OpenPCDet [20], to obtain their performance on the VANJEE PointCloud dataset. In a tunnel scene, for each of those methods, final detection results are generated utilising a result-level backend stitching strategy.

From *Table 1*, it can be observed that our algorithm outperforms others in terms of precision and recall metrics for various classes of objects. Our model outperforms SECOND by 3.0 percentage points in the person category, 4.6 percentage points in the bicycle category, 2.4 percentage points in the bus category, 3.2 percentage points in the car category and 3.1 percentage points in the truck category. Additionally, it remains competitive with two-stage algorithms. As shown in *Figure 11*, the prediction results based on the fusion of dual lidar data demonstrate that the algorithm proposed in this paper can improve the point cloud density and detection performance of distant objects in lidar.

Table 1 – 3D detection average precision (AP in %) on VANJEE PointCloud dataset

Algorithms	mAP	Car3	Bicycle	Bus	Tricycle	Person	Truck
PointPillars [12]	76.8	90.8	74.9	88.5	71.7	72.4	89.3
SECOND [10]	76.5	89.9	77.7	92.7	70.4	74.6	91.4
CenterPoint [21]	78.3	90.9	78.5	91.5	72.6	74.2	90.3
Voxel R-CNN [22]	81.6	91.7	78.9	92.8	77.2	75.8	89.2
SE-SSD [17]	82.5	91.2	79.5	93.3	78.2	76.8	92.2
PV-RCNN++ [23]	82.1	90.7	79.8	93.2	77.6	75.9	91.3
CenterFormer [24]	83.2	92.3	80.9	94.3	78.5	77.1	93.8
Ours	**83.5**	**93.1**	**81.7**	**95.1**	**78.6**	**77.6**	**94.5**

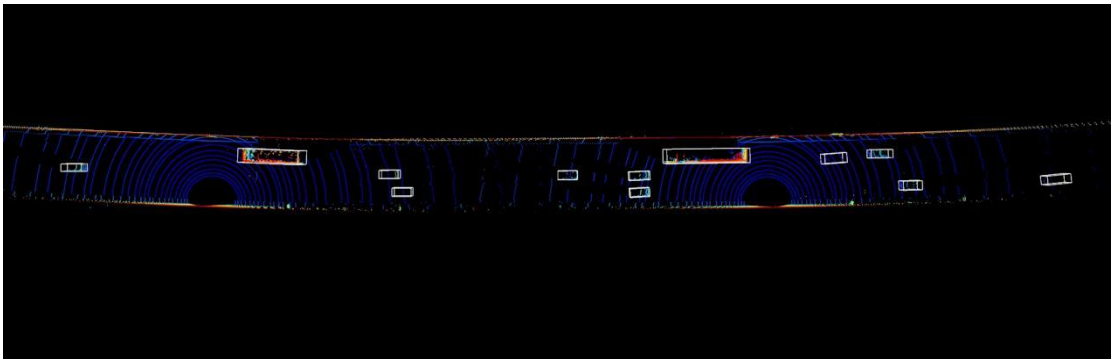


Figure 11 – Display of lidar detection results

4.4 Ablation study

We present ablation studies to analyse the effectiveness of our proposed modules. *Table 2* summarises the ablation results on our data-level point cloud stitch (DLS), "hard database" sampling (HDS), DAFA module (DAFA) and CD-IoU loss (ODIoU). For CD-IoU loss, we replace it with the smooth-L1 loss in this ablation study. All reported APs have 40 recall points. We choose SECOND as the baseline for training.

Effect of data-level point cloud stitching. Using 3,000 frames of tunnel data as the test dataset, the same point cloud detection model is used to perform target detection on the data before and after data fusion. According to *Table 2*, compared with the baseline model, it can be observed that after data fusion that the

average precision (AP) for car, bus and truck categories has significantly improved, with increases of 0.8 percentage points, 1.1 percentage points and 1.4 percentage points, respectively. However, there was no improvement in the AP for the person and bicycle categories, as their small target detection ranges are close, and the fusion of distant data has no effect on them.

We also conducted separate tests on the trajectory fusion rates before and after data fusion using different algorithms, as shown in *Table 3*.

It can be seen that compared to the global result stitch used before, no matter what kind of point cloud target detection algorithm is used, the use of data-level global point cloud stitch can effectively improve the global trajectory fusion rate. As shown in Table 3, the greater the number of lidar mosaics, the higher the global trajectory fusion rate, demonstrating the effectiveness of the data-level stitching used in the global perception system.

Table 2 – 3D detection average precision (AP in %) of proposed method with different configurations

DLS	HDS	DFAF	CD-IoU	Car	Bicycle	Bus	Tricycle	Person	Truck
				89.9	78.1	92.1	75.4	74.6	91.4
			√	90.5	79.3	92.6	76.9	76.1	91.9
		√		91.7	80.3	93.8	77.2	76.2	93.1
	√			91.2	79.9	93.3	77.5	77.4	92.7
√				90.7	78.1	93.2	75.8	74.6	92.8
√	√			92.1	80.8	94.2	77.7	76.7	93.5
√	√	√		92.6	81.2	94.7	78.1	77.1	94.0
√	√	√	√	93.1	81.7	95.1	78.6	77.6	94.5

Table 3 – Global trajectory fusion rate experiment (in %)

Method	**PointPillars** [12]	**SECOND** [10]	**CenterPoint** [21]	**Voxel R-CNN** [22]	Ours
Before stitching	89.6	89.3	91.3	91.5	**91.7**
2 lidar stitching	89.9	90.4	91.6	**92.9**	92.5
3 lidar stitching	91.6	92.1	92.8	94.5	**94.8**
4 lidar stitching	92.7	92.4	94.7	95.5	**95.6**

Effect of "hard database" sampling. To demonstrate the effectiveness of "hard database" sampling, we compared the results before and after sampling. We trained for 100 epochs and then conducted 20 epochs of fine-tuning on the results. During this process, we employed two methods: using "hard database" sampling and not using "hard database" sampling. According to *Table 2*, compared with the baseline model, it can be observed that sampling from the hard database improves the accuracy of target detection, especially for small targets. The AP for the tricycle category increases by 2.1 percentage points, and for the person category, it increases by 2.8 percentage points.

Effect of DAFA module. In this paper, we propose the DAFA module, which adequately represents target information by extracting features at different scales, and its attention mechanism selectively integrates information from various sources. As shown in *Table 2*, compared with the baseline model, this module significantly improves the AP of all categories. For instance, the AP of the car category increased by 1.8 percentage points, the bicycle category by 2.2 percentage points, and the person category by 1.6 percentage points.

Effect of CD-IoU. To better regress the size and orientation of target boxes, we propose the CD-IoU loss. According to *Table 2*, compared with the baseline model, it can be observed that using this loss function effectively enhances the AP of targets, especially for small ones. This is because the sparse point cloud on

small targets struggles to provide sufficient information for regressing their size and orientation. Specifically, the APs of the bicycle category, tricycle category and person category increased by 1.2 percentage points, 1.5 percentage points and 1.5 percentage points, respectively.

5. CONCLUSIONS

This paper proposes a novel point cloud object detection method, with key innovations including: global data fusion and challenging target sampling strategies to improve the detection rate of targets and the global trajectory fusion rate; the introduction of the DAFA module for extracting more abundant two-dimensional features and selectively fusing feature information through an attention mechanism; and the proposal of the CD-IoU loss function to better regress the size and angle of targets. Experimental results demonstrate that our method significantly outperforms baseline algorithms, highlighting its potential to enhance the accuracy of target perception and trajectory tracking rates in global perception systems. In our research, to conduct the VANJEE PointCloud dataset, manual pose calibration takes too much time and effort. In the future, we plan to utilise automatic point cloud registration methods, including traditional and deep-learning-based methods, to improve efficiency. Additionally, future research directions include further algorithm optimisation to improve performance, as well as the application of this method to broader domains such as autonomous driving and intelligent traffic management. We believe that this research outcome will have a positive impact on the development and practical application of global perception systems, driving progress and innovation in the field of intelligent transportation.

REFERENCES

[1] Wang Y, et al. A negative binomial lindley approach considering spatiotemporal effects for modeling traffic crash frequency with excess zeros. *Accident Analysis and Prevention*. 2024;207:107741. DOI:10.1016/j.aap.2024.107741.

[2] Yang Y, Yin Y, Yuan MZ. Modeling of freeway real-time traffic crash risk based on dynamic traffic flow considering temporal effect difference. *Journal of Transportation Engineering Part A Systems*. 2023;149(7):04023063. DOI:10.1061/JTEPBS.TEENG-7717.

[3] Arnold E, et al. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems.* 2019;3782-3795. DOI:10.1109/TITS.2019.2892405.

[4] Zhou Y, et al. End-to-end multi-view fusion for 3d object detection in lidar point clouds. *arXiv*. 2019. DOI:10.48550/arXiv.1910.06528.

[5] Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017;652–660. DOI:10.1109/CVPR.2017.16.

[6] Wang Y, et al. Pseudolidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019;8445–8453. DOI:10.48550/arXiv.1812.07179.

[7] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *In Advances in Neural Information Processing Systems*. 2017;30. DOI:10.48550/arXiv.1706.02413.

[8] Thomas H, et al. Kpconv: Flexible and deformable convolution for point clouds. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019;6411–6420. DOI:10.1109/ICCV.2019.00651.

[9] Ku J, et al. Joint 3d proposal generation and object detection from view aggregation. 2017. DOI:10.48550/arXiv.1712.02294.

[10] Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection. *Sensors.* 2018;18(10). DOI:10.3390/s18103337.

[11] Zhang C, Luo W, Urtasun R. Efficient convolutions for real-time semantic segmentation of 3D point clouds. *In 2018 International Conference on 3D Vision (3DV).* 2018;399–408. DOI:10.1109/3DV.2018.00053.

[12] Lang AH, et al. Pointpillars: Fast encoders for object detection from point clouds. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019;12697–12705. DOI:10.1109/CVPR.2019.01298.

[13] Yang B, Luo W, Urtasun R. Pixor: Real-time 3D object detection from point clouds. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018;7652–7660. DOI:10.1109/CVPR.2018.00798.

[14] Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud-based 3D object detection. 2017; DOI:10.48550/arXiv.1711.06396.

[15] Qi CR, et al. Frustum pointnets for 3D object detection from RGB-D data. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018;918–927.

[16] Uy MA, Lee GH. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018;4470–4479. DOI:10.1109/CVPR.2018.00470.

[17] Zhong Y, et al. SE-SSD: Self-ensembling single-stage object detector from point cloud. *In Proceedings of the IEEE Robotics and Automation Letters (RA-L)*. 2021. DOI:10.1109/CVPR46437.2021.01426.

[18] Li E, et al. SUSTech POINTS: A portable 3D point cloud interactive annotation platform system. *IEEE.* 2020. DOI:10.1109/IV47402.2020.9304562.

[19] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *In International Conference on Machine Learning.* 2016;1050–1059.

[20] Team OD. OpenPCDet: An open-source toolbox for 3D object detection from point clouds. 2020. https://github.com/open-mmlab/OpenPCDet.

[21] Yin T, Zhou X, Krahenbuhl P. Center-based 3D object detection and tracking. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021;11784–11793.

[22] Chen S, Deng J, Li P. Voxel R-CNN: towards high performance voxel-based 3D object detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[23] Yan S, et al. PV-RCNN++: Point-voxel feature set abstraction and refinement for 3D object detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2022.

[24] Hu X, et al. CenterFormer: Fully transformers for point cloud object detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.