ISSN 1330-3651(Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240730001888
Original scientific paper

MSMFAM-VoxelNeXt: LiDAR-Camera Fusion for Highway Traffic Perception

Chunsheng ZHANG*, Bibo LIU, Changwei WANG

Abstract: Roadside perception is critical for intelligent transportation systems, but faces challenges in sensor fusion and data processing. This paper proposes an enhanced perception scheme integrating LiDAR and camera data. We introduce a multi-scale multi-feature attention module (MSMFAM) to enrich voxel features, addressing issues of voxel size and semantic information extraction. Point cloud levelling and data simulation augmentation techniques improve detection accuracy across varying sensor heights. Our fusion algorithm combines LiDAR and image results with elliptical matching for enhanced target detection and classification. Experimental results show significant improvements over baseline algorithms, with mAP increases of 2.2% in point cloud detection and 1.5% infusion results. The proposed method demonstrates potential for advancing roadside perception in intelligent transportation systems.

Keywords: fusion perception; intelligent transportation; point cloud detection; roadside perception

1 INTRODUCTION

Roadside perception provides critical support for intelligent transportation systems [1-3]. By utilizing sensors such as LiDAR, cameras, and millimeter-wave radars, roadside perception systems sense the surrounding environment. In terms of road perception capabilities, roadside perception reduces the limitations of a single vehicle's perspective, offering richer and more accurate data to support intelligent driving decisions [4]. Regarding road traffic, roadside perception enables information interaction with traffic participants and infrastructure, enhancing road efficiency and safety [5].

The accuracy of road awareness is enhanced by integrating data from multiple sensors. Currently, there are three main fusion methods: data fusion, feature fusion [6-8], and result fusion. Due to the limited computational capacity of edge devices, roadside perception systems primarily adopt result fusion methods. However, current fusion algorithms face several challenges in high-speed scenarios. First, in point cloud detection algorithms, the use of a single voxel size results in poor detection performance for both small and large targets. Second, during deployment, it is difficult to ensure that the devices are installed perfectly level, leading to poor detection of distant targets. Third, data utilization is inefficient, as data collected at different installation heights cannot be reused during training. Finally, due to time discrepancies in data collection from different sensors and the high speed of vehicles, the success rate of matching infusion algorithms is low. To address these problems, this paper improves the perception scheme based on LiDAR and camera fusion. First, a multi-scale multi-feature attention module (MSMFAM) is introduced to fuse features from smaller and larger voxels, improving voxel representation and detection performance for both small and large targets. Second, a point cloud levelling method aligns point cloud heights with the training dataset, enhancing detection accuracy for distant targets caused by non-level device installations. Third, a data simulation augmentation strategy generates point cloud patterns under current installation configurations, enriching the training dataset and improving data utilization. Finally, an elliptical matching technique uses elliptical distances for secondary matching, addressing cases where IoU-based bounding boxes fail to overlap. By leveraging the current fusion algorithm, more scientific traffic management strategies can be formulated, enhancing the accuracy of accident warnings.

This paper is structured as follows: Section 2 reviews related work on point cloud detection, image detection, and fusion algorithms. Section 3 elaborates on the fusion algorithm. Section 4 details the datasets, experimental setup, and results. Finally, Section 5 concludes the paper with a summary of the findings and a discussion of future work.

2 RELATED WORK

Our method employs a result-level fusion algorithm, primarily consisting of three modules: point cloud detection, image detection, and fusion matching. To this end, we have investigated relevant algorithms.

2.1 Point Cloud Detection

In the realm of LiDAR, perception algorithms are constantly emerging, and they can be broadly categorized into three types: voxel-based methods (e.g., SECOND [9], Voxel R-CNN [10], PV-RCNN [11], VoxelNeXt [12]), pillar-based methods (e.g., PointPillars [13], CenterPoint-Pillar [14], FastPillars [15]), and range-image methods (e.g, RangeDet [16], RSN [17]). Pillar-based algorithms are fast and can meet the accuracy requirements for simple scenarios. Range-image based detection algorithms offer high speed, but due to their reliance on depth information from a single viewpoint, they lack stereoscopic information, resulting in lower detection accuracy. Voxel-based methods, while slightly slower than 2D pseudo-image-based algorithms, offer better accuracy and are thus widely used.

2.2 Image Detection

In the field of image processing, the development of deep learning-based image detection algorithms can be traced back to the neural network research of the 1960s and 1970s. At that time, due to limitations in computational power and datasets, deep learning did not gain widespread attention. It was not until 2012, when Alex Krizhevsky and

his colleagues proposed the convolutional neural network (CNN) model AlexNet, that a significant breakthrough was made in the ImageNet [18] image recognition challenge, sparking a new wave of interest in deep learning. AlexNet, with its use of GPU acceleration, deep network architecture, and techniques such as Dropout, achieved a low error rate, establishing deep learning as a key technology in image processing. Subsequently, researchers proposed various improvements and extensions, laying the foundation for the development of deep learning image detection algorithms. One of the first methods to apply deep learning to object detection was the Region-based Convolutional Neural Network (R-CNN) [19]. R-CNN used the Selective Search algorithm to generate candidate regions and then fed these regions into a CNN for classification and localization, significantly improving detection accuracy. Following this, Fast R-CNN [20], Faster R-CNN [21], and Mask R-CNN [22] were proposed, continually enhancing the speed and accuracy of object detection algorithms. These algorithms introduced innovations such as the Region Proposal Network (RPN) and shared convolutional features, achieving faster and more precise object detection.

Meanwhile, single-stage object detection algorithms have developed more rapidly, starting with YOLO (You Only Look Once) [23] and SSD (Single Shot MultiBox Detector) [24] algorithms, undergoing rapid iterations. The introduction of Focal Loss in RetinaNet [25] has significantly improved the accuracy of single-stage algorithms, making their precision approach that of two-stage algorithms. CenterNet [26] uses an anchor-free method for object detection, avoiding the need to manually set anchor sizes. Currently, the YOLO series algorithms are continuously being updated and iterated, making them the mainstream object detection algorithms. Due to their high accuracy, low resource consumption, and fast inference speed, they are widely used in both academia and industry.

2.3 Fusion Matching

Caltagirone et al. utilized deep learning to propose a road detection algorithm based on the fusion of LiDAR point clouds and camera images. Their approach involves projecting unstructured, sparse point clouds onto the camera image plane and then sampling the projected plane, which contains both point cloud and image data, to obtain a dense 2D image that encodes spatial information, thus achieving road segmentation [27]. This method adds depth information to images, but the sparsity of point clouds leaves many areas without projections, resulting in suboptimal features. Zhijian Liu et al. proposed the BEVFusion algorithm, which fuses sensor features on the BEV plane for more accurate integration. However, it relies on synchronized multi-sensor data for training, limiting the use of single-sensor data. Cui L, Li, et al. proposed the MMFusion method [28], which uses a point cloud detection branch and an image detection branch to process data from different sensors separately.

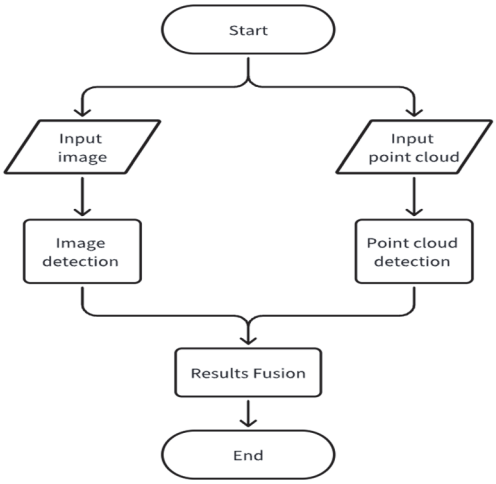

Figure 1 Flowchart of the fusion algorithm. The algorithm inputs image and point cloud, using their respective perception algorithms for detection. Then, the fusion algorithm processes the perception results to output the final result

Figure 2 Diagram of the point cloud detection algorithm workflow (training phase)

It performs fusion at the output feature level to reduce fusion parameters but requires large amounts of data to learn the relationships between sensor outputs.

This paper uses result-level fusion, adopting a VoxelNeXt-like point cloud detection algorithm with the MSMFAM module for feature extraction. YOLOv5 handles image detection, while IoU and elliptical matching replace network-based target association in fusion.

3 METHOD

The fusion perception scheme proposed in this paper mainly includes a point cloud detection algorithm, an image detection algorithm, and a result matching fusion algorithm. The point cloud detection algorithm, MSMFAM-VoxelNeXt, has been modified in certain network structures to reduce computational load. To improve the detection performance for small and large targets, an attention mechanism was added to the voxel feature extraction process. At the data level, data leveling was performed to make the model's position and size regression more accurate. Data from other installation methods were simulated to generate point clouds for the current installation method, enhancing target training. For image detection, the YOLOv5 algorithm was selected. In the result matching fusion algorithm, IoU (Intersection over Union) and elliptical matching were used for secondary association to improve fusion success rates. For the fusion results, the image target's category was assigned to the point cloud target, and the confidence was selected as the maximum value between the confidence levels of the image and point cloud detection results, resulting in more accurate target detection. The overall process is illustrated in Fig. 1.

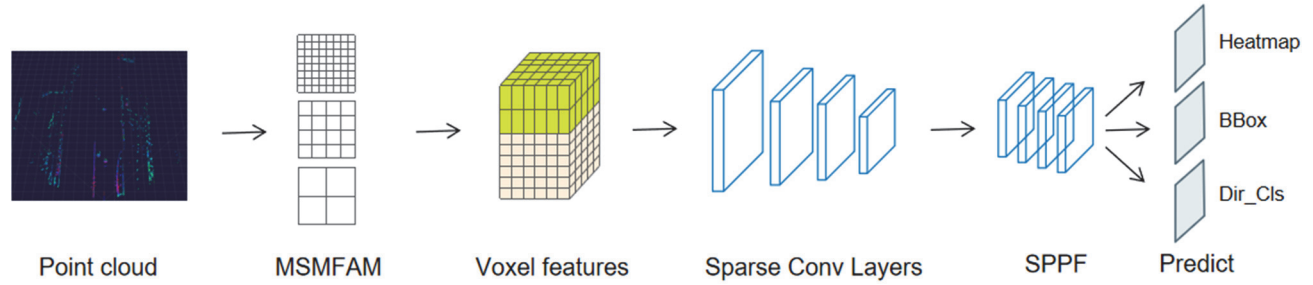

Figure 3 The overall architecture of MSMFAM-VoxelNeXt is as follows: First, the MSMFAM module is used to partition the point cloud data, extract multi-scale voxel features, and perform feature fusion. Next, 3D sparse convolution is applied to further refine the voxel features. Then, the SPPF module is used for down-sampling and up-sampling. Finally, a prediction head is employed to predict the targets

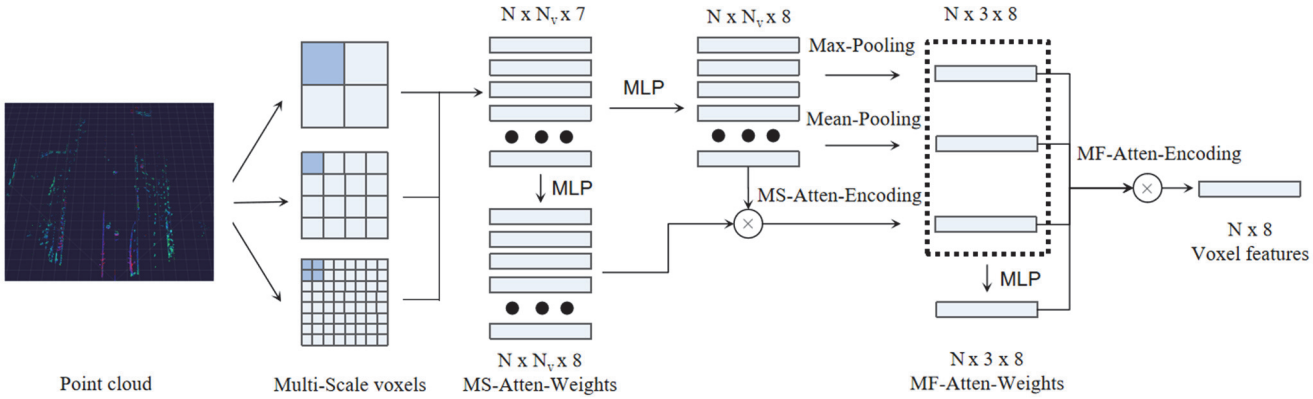

Figure 4 The overview of MSMFAM. The voxel features at different scales on the left side are processed through feature extraction and a multi-scale attention mechanism to obtain voxel attention features. On the right side, a multi-feature attention mechanism is used to weight different features, resulting in the final output features

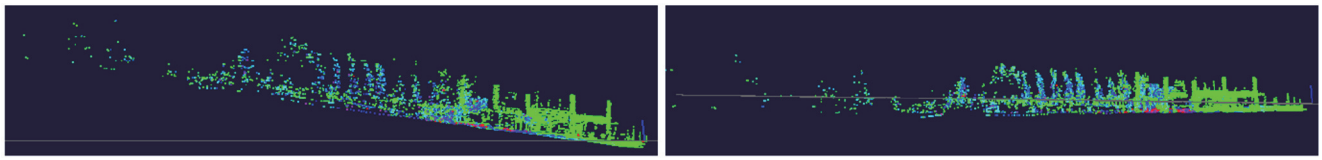

Figure 5 Side view of point cloud data. (a) Original point cloud side view. (b) Point cloud side view after leveling

3.1 Point Cloud Object Detection

In this paper, we mainly optimized the point cloud algorithm and performed data leveling and simulation augmentation. The workflow during the training process is shown in Fig. 2.

MSMFAM-VoxelNeXt: VoxelNeXt is a novel point cloud detection algorithm that uses sparse convolution throughout its network to avoid generating features in empty regions. This reduces computational load and memory usage while supporting long-range target detection. It is highly accurate among point cloud methods and produces precise fusion results, making it ideal for roadside environmental monitoring. However, there are still some issues within the network structure. The feature representation capability provided by early voxels is limited, representing only the information within the current voxel size. Additionally, due to the fully sparse convolutional network architecture, more down-sampling layers are needed later to increase the receptive field.

To address these two issues and to ensure compatibility with platforms that have lower computational power, we made some improvements to VoxelNeXt. This improved algorithm is named MSMFAM-VoxelNeXt. The overall architecture of the algorithm is shown in Fig. 3.

Regarding the semantics represented by voxels, the current feature extraction method using averaging is overly simplistic. Smaller voxel division sizes provide more accurate positional information but result in exponentially increased computational load, necessitating more feature extraction layers to obtain high-level semantics. Conversely, larger voxel division sizes reduce computational load and provide higher semantic information, but result in blurred object contours, leading to poorer detection and localization performance. To address this issue, we propose an effective attention module (MSMFAM).

To better fuse different input features, we utilize an attention mechanism to predict the weights of different input features. Existing attention mechanisms, such as SENet [29] and CBAM [30], are designed for a fixed number of input features. If the number of input features changes or the feature distribution varies, it can impact the weight prediction. However, due to the sparsity of point clouds, not all locations contain points. To address this, we optimized the attention mechanism by using an MLP to predict the weights for each feature. This approach avoids the impact of varying input feature numbers on weight prediction. For feature information merging, we adopt a

method similar to the pillar approach. After using MLP for feature extraction, we perform max pooling to obtain the features. However, using only max pooled features may be too simplistic. Therefore, we use an attention mechanism to generate weights for different features, and then perform a weighted sum to obtain the attention features. For all features, we calculate the mean along the corresponding dimensions to obtain the mean pooled features. Based on these three different features, we apply another attention mechanism to calculate the corresponding weights for the different features, resulting in the final voxel input features. These features contain both fine positional information and local contour information, significantly improving the detection of small and easily confused objects. The entire process is illustrated in Fig. 4. By default, the voxel's length and width are set to 0.1 m, and the height is chosen to generate voxel features according to the aforementioned scheme. Additionally, the voxel sizes are halved and doubled, respectively, to obtain voxel features at three scales. These features are then fed into the MSMFAM to obtain the baseline voxel features.

We utilize a voxelization technique that involves unequal division in height, where the region containing the target is divided more densely, and the upper region of the target is divided into larger sizes. The dense division in the target region helps to distinguish between targets with similar shapes. The larger scale division in the upper detection region is used only for height prediction.

Data Optimization: To further enhance the performance of the MSMFAM-VoxelNeXt algorithm, several data optimization techniques are employed. These techniques aim to make the predictions more accurately reflect the true positions and sizes of the targets and increase the utility of the point cloud data. They are data leveling and point cloud simulation augmentation.

By adjusting the point cloud to a horizontal level, the issue of excessive heights for distant point clouds exceeding the detection range can be avoided. Additionally, aligning the input height with the training data improves the accuracy of target detection. We adjust the angles of the point clouds through an automated method. The results are shown in Fig. 5. The process is as follows:

Step 1: Calculation of the Ground Normal Vector in the Point Cloud Coordinate System.

We assume the ground is a plane, described by the following equation. Using the RANSAC algorithm, three points are randomly selected from the point cloud in each iteration to calculate the plane's coefficients. The plane with the most points close to it is chosen as the final result, and its coefficients represent the ground's normal vector.

$$A \cdot x \cdot B \cdot y + C \cdot z + D = 0 \tag{1}$$

Step 2: Calculation of the Deflection Angle Based on the Ground Normal Vector.

To calculate the angular difference between the normal vector of the fitted plane from the point cloud and the true normal vector along the three axes, we obtain the pitch and roll angles. Suppose the normal vector of the ground plane calculated from the point cloud is $norm_1(a_1, b_1, c_1)$, and the true ground normal vector is (0, 0, 1). θ_x and θ_y are the radians rotated around the x and y axes.

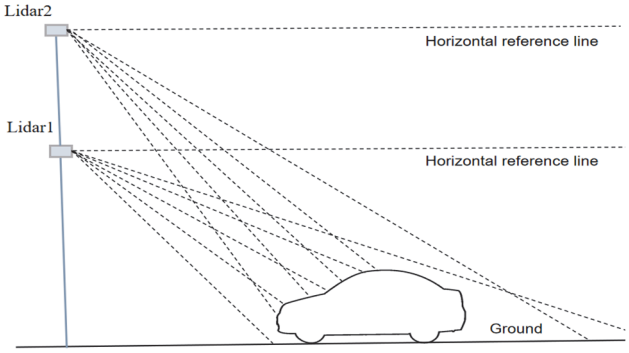

Figure 6 Schematic diagram of target point cloud harness distribution at different installation method

The relationship between them is given by:

$$\begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} \sin\theta_y \\ -\sin\theta_x \cos\theta_y \\ \cos\theta_x \cos\theta_y \end{bmatrix} \tag{2}$$

$$\theta_x = \arcsin\left(\frac{-b}{\cos\theta_y}\right) \tag{3}$$

$$\theta_y = \arcsin\left(a_1\right) \tag{4}$$

Therefore, given the point cloud's rotation angles θ_x around the x-axis and θ_y around the y-axis, we can level the point cloud by rotating it in the reverse direction. First, rotate the points clockwise by θ_x around the x-axis, followed by a rotation by θ_y around the y-axis. This results in a leveled point cloud.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5}$$

In different scenarios, roadside perception equipment may be installed at varying methods (as shown in Fig. 6). Data from different scenarios is difficult to reuse directly, so traditional methods require creating a dataset for each scenario for model training and evaluation. Each dataset must contain at least tens of thousands of frames, and the creation process includes data collection, screening, annotation, and quality inspection, which is time-consuming and significantly delays project progress. To solve this problem, this paper proposes a data processing method that is applicable to different heights and angles. Other dataset is used to enhance the training effect of current dataset as Algorithm 1. The data simulation augmentation method extracts the category and position of data from datasets of other installation methods. It simulates the scanning process of the current LiDAR installation configuration on the target position. The target is selected from a library of high-precision geometric models based on its category and is placed into the scene using the calibrated position and heading angle. Finally, the LiDAR scans according to horizontal and vertical angles,

calculating the intersection points with the high-precision geometric model to obtain the contour of the target.

Algorithm 1 *Data Simulation Augmentation with Multiple Geometric Models*

Input: *Dataset 1 (D1), Current dataset; Dataset 2 (D2), Other installation method dataset; High-Precision Model Library (HPM) with multiple models per category*
Output: *Augmented Dataset 1 (D1)*
1:Initialize empty object database ODB
2:Load multiple high-precision geometric models foreach category into HPM
3: ***for*** *each object A in D2* ***do***
4:Extract bounding box B_A and category C_A of object A
5: Select a corresponding geometric model M_A from HPM based on C_A(e.g., randomly choose or use heuristicselection)
6: Place M_A at the position and orientation defined by B_A
7: Translate M_A to new height H, obtaining adjusted model M_B
8: Simulate all points that lidaratheight H can hit on M_B based on lidar specifications
9: Store simulated points P_B and metadata {B_A , M_B , difficulty level(B_A)} in ODB
10: ***end for***
11: ***for*** *each frame F in D1* ***do***
12: Detect road areas in F to obtain road areas
*13: **for** each object in ODB**do***
14: Select objects from ODB for augmentation
15: Determine reasonableplacement positionwithinroad areas
16: Place selected objects in F at determinedpositions
*17: **end for***
18: ***endfor***

3.2 Image Object Detection

The image detection algorithm is parallel to the point cloud detection algorithm, processing sensor data to predict targets and output 2D bounding boxes, providing foundational data for the fusion algorithm. This paper employs YOLOv5, a deep learning-based object detection algorithm by the ultralytics team. YOLOv5 enhances speed and accuracy over its predecessors in the YOLO series.

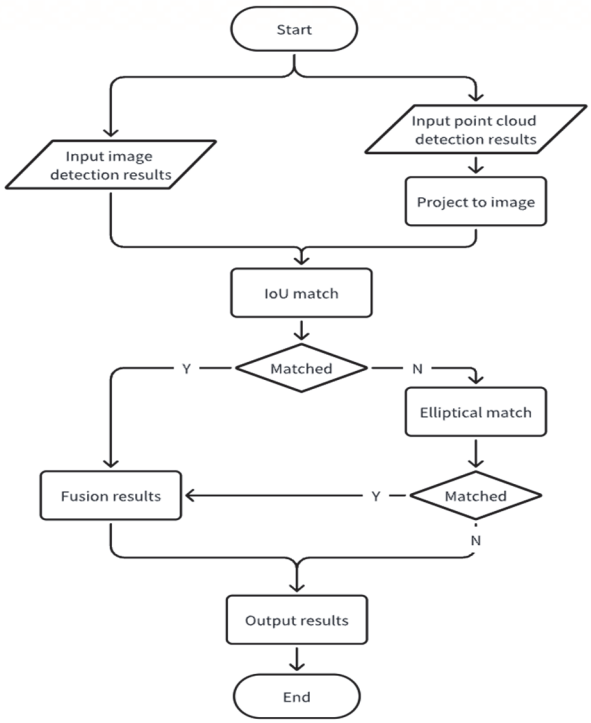

Figure 7 Flowchart of the results fusion algorithm. Perform two matches for different sensors, and finally output the results

Here are some features and advantages of the YOLOv5 algorithm:
1) Lightweight design: The network structure of YOLOv5 has been simplified and optimized, achieving faster inference speed and higher detection accuracy.
2) Multi-scale detection: YOLOv5 performs object detection on feature maps at different levels, effectively detecting objects of various sizes.
3) Data augmentation: By using data augmentation techniques, YOLOv5 can expand the dataset during the training phase, improving the model's generalization ability.
1) Adaptive training: YOLOv5 supports adaptive training, allowing flexible adjustments based on different hardware environments and task requirements.

Overall, YOLOv5 is an efficient and accurate object detection algorithm suitable for various computer vision tasks, such as object detection, pedestrian detection, and more.

3.3 Results Fusion

In result-level fusion, we project 3D point cloud bounding boxes onto 2D images and perform result fusion with the detection boxes in the 2D image. This significantly reduces the complexity of the algorithm and is more conducive to data association. The matching is primarily based on the image boxes, with the IOU threshold set to match the point cloud bounding boxes. For objects that do not intersect, an elliptical matching is performed based on category information for secondary association. When generating the 2D elliptical equation, the target's height is used as b and width as a, with the center point as the ellipse's center, as shown in Eq. (6). If the point cloud's projection onto the image satisfies the equation (≤ 1), fusion is possible. If multiple targets match, the one with the smallest distance is chosen as the final result.

$$distance_e = \frac{(x-x_c)^2}{w^2} + \frac{(y-y_c)^2}{h^2} \tag{6}$$

This ultimately achieves the deduplication and information fusion of different sensors. The specific flowchart is shown in Fig. 7. First, the results of the image detection algorithm and point cloud detection algorithm are input, and the point cloud detection results are projected onto the image plane. Next, the IoU between the point cloud bounding boxes and image bounding boxes is calculated. If the IoU is below the threshold, elliptical matching is performed. Finally, targets with an IoU above the threshold and successfully matched through elliptical matching are fused to obtain the final results. If the object identity binding categories are inconsistent, the visual category takes precedence. Images are rich in color and dense texture information, which leads to better performance in category accuracy. By using a fusion algorithm to assign image detection categories to point cloud detection categories, the misclassification of different categories with similar shapes and sizes in point cloud detection can be reduced. For the confidence of the fusion target, the highest confidence between the image and point cloud detection results is selected as the final

confidence. By enhancing the fusion target confidence, false detections can be better distinguished from true targets.

4 EXPERIMENTS

We present our experimental results on a self-created dataset, and finally test on a single dataset. Experimental results verify that the aforementioned improvements are effective.

4.1 Datasets

This study utilized two datasets. They are captured by a 10 FPS LiDAR sensor with 64 lines in 360°. They provide bounding box annotations for five classes: pedestrian, car, bus, truck, and motorcycle. The evaluation metric is mean average precision (mAP).

Dataset 1, collected with sensors mounted at a height of 6m, is ideal for highway and tunnel scenarios. It contains 18000 annotated frames of images and point clouds, divided for training, validation, and testing of visual and point cloud algorithms. The train set consists of 12000 frames, the validation set consists of 3000 frames, and the test set consists of 3000 frames.

Dataset 2 has its perception equipment installed at a height of 4 m, making it suitable for highway scenarios. It comprises a total of 7000 frames of data (both images and point clouds, all fully annotated). The train set consists of 5000 frames, the validation set consists of 1000 frames, and the test set consists of 1000 frames. Only the data simulation augmentation ablation experiment was tested on both datasets; other results were based on tests conducted using Dataset 1.

The distribution of each category in the dataset is shown in Fig. 8. As it is a highway dataset, the data exhibits a noticeable long-tail effect, with the car category being the most abundant, while other categories are relatively sparse. Compared to datasets with balanced category distributions, detecting minority target categories is more challenging and may result in poorer performance, especially for some difficult-to-detect categories.

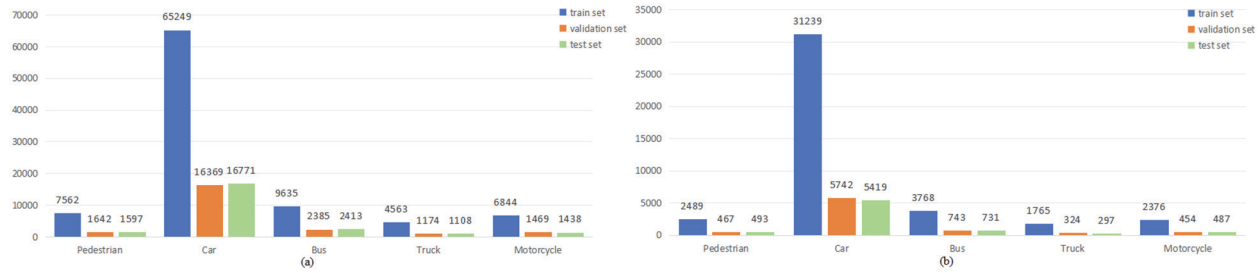

Figure 8 Dataset distribution: the left side represents Dataset 1, and the right side represents Dataset 2

4.2 Implementation Details

We set the range of 3D voxel space as [−100.8 m, 100.8 m] for the X and Y axes, and [−6 m, 0 m] for the Z axis. In other algorithms, the voxel size is set to (0.1 m, 0.1 m, 0.15 m). In the MSMFAM-VoxelNeXt, the baseline voxel size in the X and Y axes is set to (0.1 m, 0.1 m). The voxel height is set to 0.1 m in the range of −6 m to −3 m, and 0.3 m in the range of −3 m to 0 m. In the X and Y axes directions, the dimensions of the other two voxelization methods are halved and doubled, respectively, while the Z axis dimensions remain the same as the current setting. These parameters are set for Dataset 1. The parameters for Dataset 2 are set as follows: the height range is [−4 m, 0 m], and the voxel size in the X and Y directions is the same as previously set. The voxel height is set to 0.075 m in the range of [−4 m, −1.75 m], and 0.175 m in the range of [−1.75 m, 0 m].

We train the network with the batch size 32, learning rate 0.03 for 100 epochs on 8 Tesla V100S GPUs using the Adam optimizer. We select the model from the last epoch of training for testing.

4.3 Object Detection Results

In Tab. 1, we compare the accuracy of the MSMFAM-VoxelNeXt with other classical point cloud algorithms. Compared to the baseline algorithm VoxelNeXt, our proposed algorithm shows a significant improvement. Our model outperforms VoxelNeXt by 2.1% in the pedestrian category, 2.6% in the motorcycle category, 2.1% in the bus category, 1.6% in the car category, 2.3% in the truck category, and 2.2% in the mAP. Additionally, it remains competitive with two-stage algorithms. This is attributed to MSMFAM and data simulation augmentation.

The inference time and FLOPs of the optimized detection algorithm are shown in Tab. 2. As we can see, compared to the baseline algorithm, the inference time only increases by 5 ms, and the FLOPs increase by 2.3 G, while the mAP improves by 3.3%. The algorithm has low complexity and minimal time increase, making it suitable for real-time detection applications.

In Tab. 3, we compare the accuracy of using only point cloud data versus the fusion scheme, demonstrating that our proposed method is highly effective, especially with significant improvements for pedestrian and motorcycle categories. This is mainly because the fusion algorithm can correct classification errors predicted in the point cloud and improve the confidence of successfully fused targets. Due to the smaller size of pedestrian and motorcycle targets and the limited number of points in the point cloud, the detection confidence is very low. Additionally, the similarity between these targets and the background increases the likelihood of misclassification between categories. On the other hand, image detection algorithms, benefiting from richer texture and color information, perform better in detecting pedestrians and motorcycles with higher confidence levels. By employing the fusion

algorithm, the reliability of target confidence and classification can be improved, thereby enhancing overall detection accuracy.

Table 1 Detection Results of Different Point Cloud Detection Algorithms on the AP Metric (%)

Method	Pedestrian	Car	Bus	Truck	Motorcycle	mAP
SECOND	81.3	90.2	85.7	84.6	78.2	84
PointPillar	79.8	89.1	85.3	84.5	76.5	83
CenterPoint	82.7	91.2	86.3	85.1	79.1	84.9
VoxelRCNN	83.2	91.4	86.9	86.3	79.8	85.5
VoxelNeXt	83.4	91.6	86.8	86.1	80.3	85.6
OURS	85.5	93.2	88.9	88.4	82.9	87.8

Table 2 The Test Results of Algorithm Inference Time and FLOPs

Model	Latency / ms	FLOPs / G	mAP
Baseline	59	82.4	84.5
OURS	64	84.7	87.8

Table 3 Comparison of LiDAR-Only and Fusion Algorithms Results on the AP Metric (%)

Model	Pedestrian	Car	Bus	Truck	Motorcycle	mAP
LiDAR	85.5	93.2	88.9	88.4	82.9	87.8
Fusion	88.2	94.5	90.4	89.2	84.3	89.3

By using the current fusion algorithm, higher accuracy can provide more precise information for traffic decision-making. Adjusting the point cloud horizontally can reduce the requirements and complexity of radar installation. Data simulation enables rapid data generation, allowing the algorithm to adapt to new scenarios and improve performance outcomes.

4.4 Detection Performance

In this section, we will showcase the detection performance of our final model on images, point clouds, and fusion results. The left image shows the left-side view of the point cloud, the middle image shows the right-side view of the point cloud, and the right image presents the detection results of the point cloud algorithm. Blue boxes indicate cases where the image results and point cloud results did not match successfully, while green boxes indicate successful matches between the image results and point cloud results. In Fig. 9, there are detection results of vehicles at different distances, detection results of large-sized targets (bus) at long distances, and detection results of targets in occlusion scenarios. Our algorithm is able to detect all scenarios stably.

Since the current fusion algorithm primarily addresses the issue of unmatched bounding boxes in high-speed scenarios, it may not be applicable to low-speed urban scenarios. Although the current algorithm improves the detection performance for pedestrians and motorcycles, its performance is still inferior to other categories. In the future, we will focus on studying fusion algorithms for multi-sensor and multi-scenario applications, aiming to enhance the accuracy and generalizability of the algorithms.

4.5 Ablation Study

In this section, we conduct extensive ablation experiments to analyze individual components of our proposed method.

Effects of MSMFAM. We proposed an efficient attention encoding method called MSMFAM, which can be applied to any voxel-based method. MSMFAM uses smaller voxels to extract local information, providing more small-scale features for fusion, which improves detection performance for small targets such as pedestrians and motorcycles. By utilizing larger voxel sizes for feature extraction, the receptive field is expanded, enhancing the model's detection performance for large targets, with significant improvements observed in the bus and truck categories. Additionally, the attention mechanism enables selective extraction of information from different sources. As shown in the 2nd and 3rd rows of Tab. 4, the accuracy for the pedestrian category increased by 1.6%, for the motorcycle category by 1.9%, for the bus category by 1.3%, and for the truck category by 1.2%.

Effects of SPPF. To verify the effectiveness of SPPF, we selected VoxelNeXt as the baseline model and removed the additional sparse convolution stage added in the paper. On this basis, we added the SPPF module. Thanks to the larger pooling kernel used in SPPF, the detection performance for large targets was slightly improved. As shown in Tab. 4, the mAP increased by 1.2%, with the Bus and Truck categories both improving by 2.2%.

Effects of leveling point clouds. Point cloud leveling can prevent the accuracy degradation caused by inconsistencies between the point cloud distribution and the training distribution due to the LiDAR not being installed horizontally. When an angle exists, as the distance increases, the target deviates more significantly from the original training height and may even exceed the detection range. The annotation distance for general vehicles is usually farther, so the accuracy improvement for vehicles tends to be more significant. As shown in the third and fourth rows of Tab. 4, the mAP increased by 0.3%, and the accuracy for vehicles improved by 0.5%.

Effects of data simulation augmentation. We propose a data simulation augmentation method to improve data utilization. By using point cloud objects at different heights to simulate the current height for data augmentation, detection accuracy can be enhanced. As shown in the 4th and 5th rows of Tab. 4, the mAP increased by 0.4%. In the Tab. 5, the mAP increased by 0.6%.

Table 4 Comparison of Different Configurations on the AP Metric (%)

MSMFAM	SPPF	Leveled data	Data augmentation	Fusion	Elliptical matching	Pedestrian	Car	Bus	Truck	Motorcycle	mAP	ΔmAP
						82.7	90.9	84.9	84.3	79.6	84.5	0
	√					83.2	91.5	87.1	86.5	80.4	85.7	+1.2
√	√					84.8	92.4	88.4	87.7	82.3	87.1	+1.4
√	√	√				85	92.9	88.7	88.2	82.4	87.4	+0.3
√	√	√	√			85.5	93.2	88.9	88.4	82.9	87.8	+0.4
√	√	√	√	√		86.7	94.1	89.5	88.7	83.7	88.5	+0.7
√	√	√	√	√	√	88.2	94.5	90.4	89.2	84.3	89.3	+0.8

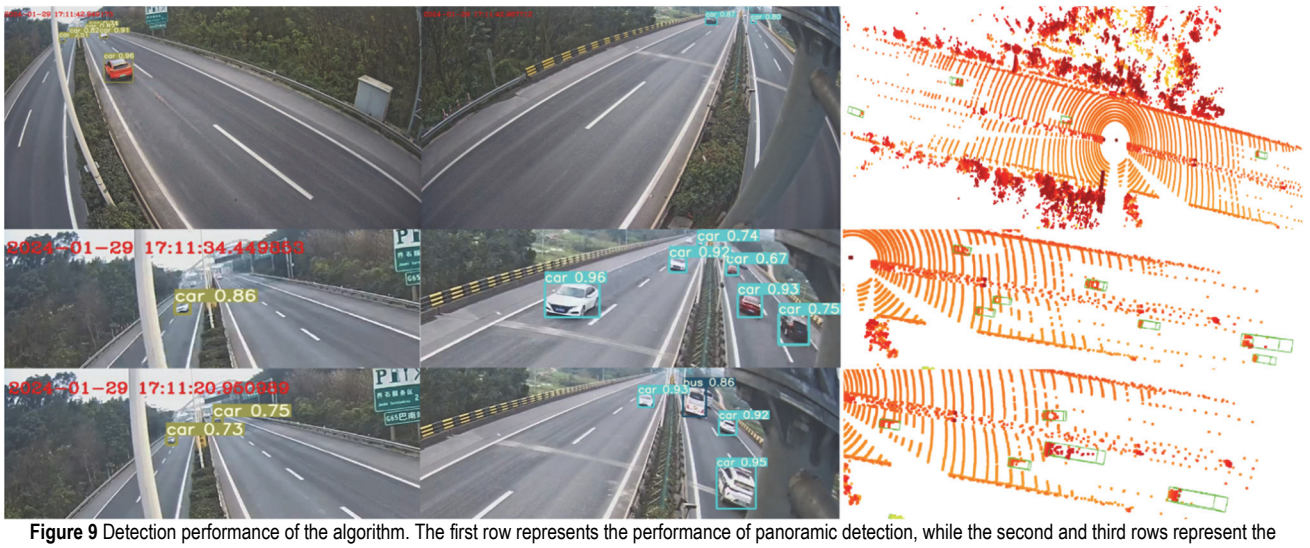

Figure 9 Detection performance of the algorithm. The first row represents the performance of panoramic detection, while the second and third rows represent the performance of local field detection

Effects of Fusion and Elliptical matching. To fully utilize the advantages of images, we proposed a fusion module for point cloud and image detection. Image detection improves accuracy and confidence for categories like pedestrians and motorcycles due to rich texture details, while point cloud detection offers higher localization accuracy. Using image-based category correction and maximum confidence for matched targets ensures more stable results. As shown in Tab. 4, the fusion algorithm improved mAP by 0.7% (pedestrians +1.2%, motorcycles +0.8%), and adding elliptical matching further boosted mAP by 0.8%.

Table 5 Comparison of Using Simulated Data Augmentation on Training and Testing on Dataset 2 (%)

augmentation	Pedestrian	Car	Bus	Truck	Motorcycle	mAP
normal	82.1	90.5	85.2	85.4	82.3	85.1
simulated	83.2	90.9	85.7	85.9	83	85.7

5 CONCLUSIONS

This paper presents an enhanced perception scheme for intelligent transportation systems, integrating LiDAR and camera data for roadside perception tasks. Our MSMFAM-VoxelNeXt algorithm, combined with data optimization techniques and a novel fusion approach, demonstrates significant improvements in object detection accuracy and efficiency. The proposed method outperforms baseline algorithms, with mAP increases of 2.2% in point cloud detection and 1.5% in fusion results. These improvements are particularly notable for challenging categories such as pedestrians and motorcycles. Our approach addresses key challenges in roadside perception, including varying sensor heights and fusion accuracy. Future work should focus on real-time implementation and testing in diverse traffic scenarios. This study enhances the accuracy and reliability of roadside perception, providing reliable support for autonomous driving, improving traffic efficiency, and contributing to the advancement of intelligent transportation systems. In the future, we will integrate results from more sensors to improve the detection of hard-to-detect targets.

6 REFERENCES

[1] Wang, W., Yang, Y., Yang, X., Gayah, V. V., Wang, Y., Tang, J., & Yuan, Z. (2024). A negative binomial Lindley approach considering spatiotemporal effects for modeling traffic crash frequency with excess zeros. *Accident Analysis & Prevention*, *207*, 107741. https://doi.org/10.1016/j.aap.2024.107741

[2] Yang, Y., Yin, Y., Wang, Y., Meng, R., & Yuan, Z. (2023). Modeling of freeway real-time traffic crash risk based on dynamic traffic flow considering temporal effect difference. *Journal of transportation engineering, Part A: Systems*, *149*(7), 04023063. https://doi.org/10.1061/JTEPBS.TEENG-7717

[3] Yang, Y., Yang, B., Yuan, Z., Meng, R., & Wang, Y. (2024). Modelling and comparing two modes of sharing parking spots at residential area: Real-time and fixed-time allocation. *IET Intelligent Transport Systems*, *18*(4), 599-618. https://doi.org/10.1049/itr2.12343

[4] Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., ... & Nie, Z. (2022). Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21361-21370. https://doi.org/10.1109/CVPR52688.2022.02067

[5] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652-660.

[6] Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., & Tang, Z. (2022). Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, *35*, 10421-10434.

[7] Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D. L., & Han, S. (2023). Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *2023 IEEE international conference on robotics and automation (ICRA)*, 2774-2781. https://doi.org/10.1109/ICRA48891.2023.10160968

[8] Hu, H., Wang, F., Su, J., Wang, Y., Hu, L., Fang, W., & Zhang, Z. (2023). Ea-lss: Edge-aware lift-splat-shot framework for 3d bev object detection. *arXiv preprint arXiv:2303.17895*.

[9] Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337. https://doi.org/10.3390/s18103337

[10] Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., & Li, H. (2021). Voxel r-cnn: Towards high performance voxel-based

3d object detection. *Proceedings of the AAAI conference on artificial intelligence*, *35*(2), 1201-1209. https://doi.org/10.1609/aaai.v35i2.16207

[11] Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10529-10538. https://doi.org/10.1109/CVPR42600.2020.01054

[12] Chen, Y., Liu, J., Zhang, X., Qi, X., & Jia, J. (2023). Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21674-21683. https://doi.org/10.1109/CVPR52729.2023.02076

[13] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697-12705. https://doi.org/10.1109/CVPR.2019.01298

[14] Yin, T., Zhou, X., & Krahenbuhl, P. (2021). Center-based 3d object detection and tracking. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784-11793. https://doi.org/10.1109/CVPR46437.2021.01161

[15] Zhou, S., Tian, Z., Chu, X., Zhang, X., Zhang, B., Lu, X., & Ma, L. (2023). Fastpillars: A deployment-friendly pillar-based 3d detector. *arXiv preprint arXiv:2302.02367*, *9*.

[16] Fan, L., Xiong, X., Wang, F., Wang, N., & Zhang, Z. (2021). Rangedet: In defense of range view for lidar-based 3d object detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 2918-2927. https://doi.org/10.1109/ICCV48922.2021.00291

[17] Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., & Anguelov, D. (2021). Rsn: Range sparse net for efficient, accurate lidar 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5725-5734. https://doi.org/10.1109/CVPR46437.2021.00567

[18] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[19] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *38*(1), 142-158. https://doi.org/10.1109/TPAMI.2015.2437384

[20] Girshick, R. (2015). Fast r-cnn. *arXiv preprint arXiv:1504.08083*. https://doi.org/10.1109/ICCV.2015.169

[21] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[22] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961-2969. https://doi.org/10.1109/ICCV.2017.322

[23] Redmon, J. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1109/CVPR.2016.91

[24] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14*, 21-37. Springer International Publishing. https://doi.org/10.1007/978-3-319-10578-9_23

[25] Lin, T. (2017). Focal Loss for Dense Object Detection. *arXiv preprint arXiv:1708.02002*. https://doi.org/10.1109/ICCV.2017.324

[26] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 6569-6578. https://doi.org/10.1109/ICCV.2019.00667

[27] Caltagirone, L., Bellone, M., Svensson, L., & Wahde, M. (2019). LIDAR-camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, *111*, 125-131. https://doi.org/10.1016/j.robot.2018.11.002

[28] Cui, L., Li, X., Meng, M., & Mo, X. (2023, November). MMFusion: A Generalized Multi-Modal Fusion Detection Framework. *2023 IEEE International Conference on Development and Learning (ICDL)*, 415-422. https://doi.org/10.1109/ICDL55364.2023.10364441

[29] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[30] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. *Proceedings of the European conference on computer vision (ECCV)*, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1

Contact information:

Chunsheng ZHANG
(Corresponding author)
Guangdong Provincial Highway Construction Co., Ltd,
Guangzhou, Guangdong Province, China
E-mail: 77862936@qq.com

Bibo LIU
Boshen Branch of Guangdong Boda Expressway Co., Ltd,
Guangzhou, Guangdong Province, China

Changwei WANG
Boshen Branch of Guangdong Boda Expressway Co., Ltd,
Guangzhou, Guangdong Province, China