Pedestrian detection based on YOLOv3 multimodal data fusion

Cheng Wang\textsuperscript{a}, Yuan-sheng Liu\textsuperscript{b}, Fei-xiang Chang\textsuperscript{a} and Ming Lu\textsuperscript{c}

\textsuperscript{a}Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, People’s Republic of China; \textsuperscript{b}College of Robotics, Beijing Union University, Beijing, People’s Republic of China; \textsuperscript{c}College of Applied Science and Technology, Beijing Union University, Beijing, People’s Republic of China

ABSTRACT

Multi-sensor fusion has essential applications in the field of target detection. Considering the current actual demand for miniaturization of on-board computers for driverless vehicles, this paper uses the multimodal data YOLOv3 (MDY) algorithm for pedestrian detection on embedded devices. The MDY algorithm uses YOLOv3 as the basic framework to improve pedestrian detection accuracy by optimizing anchor frames and adding small target detection branches. Then the algorithm is accelerated by using TensorRT technology to improve the real-time performance in embedded devices. Finally, a hybrid fusion framework is used to fuse the LIDAR point cloud data with the improved YOLOv3 algorithm to compensate for the shortcomings of a single sensor and improve the detection accuracy while ensuring speed. The improved YOLOv3 improves \textit{AP} by 6.4\% and speed by 11.3 FPS over the original algorithm. The MDY algorithm achieves better performance on the KITTI dataset. To further verify the feasibility of the MDY algorithm, an actual test was conducted on an unmanned vehicle with Jetson TX2 embedded device as the on-board computer within the campus scenario, and the results showed that the MDY algorithm achieves 90.8\% accuracy under real-time operation and can achieve adequate detection accuracy and real-time performance on the embedded device.

1. Introduction

Multimodal data fusion is the fusion of multiple sensor data using fusion algorithms, which has higher reliability and can obtain richer target information than a single sensor (Huang et al., 2022). Therefore, fusing multiple sensors to locate pedestrian targets accurately can provide a more reliable basis for subsequent decision-making and control of unmanned vehicles, which can significantly ensure the driving safety of unmanned vehicles.

In the field of multi-sensor fusion for target detection, each sensor has its advantages and limitations (Cui et al., 2020). The accuracy with which the camera receives information from the environment depends on the amplitude and frequency of the light waves. Therefore, changes in lighting can affect the performance of the camera. However, LIDAR actively emits laser pulses to sense the environment and is slightly affected by external lighting conditions but can be affected by extreme weather. Currently, compensating for the shortage of a single sensor by fusing multiple sensors has been a hot research topic in this field.

In recent years, multi-sensor fusion technologies have emerged. Jason et al. (2018) proposed AVOD, which differs from MV3D (Chen et al., 2017) in that it improves the detection speed and accuracy by discarding the density features in the forward and bird’s-eye views of LIDAR from the original input. Deng and Czarnecki (2019) proposed MLOD, which generates 3D candidate regions from a regional network in a bird’s eye view (BEV) projection of a point cloud, and then sends the corresponding information to the detector for classification and bounding box regression using 3D proposal bounding boxes projected onto the image and BEV feature maps, so that detection performance is effectively improved. Wang and Jia (2019) proposed F-ConvNet, which uses 2D images for detection to generate a series of 2D detection frames, then generates a series of cones for local point grouping based on the detection frames, then performs point-wise feature extraction and combines them into cone-level features as input to the fully connected layer to achieve end-to-end continuous position regression. Vora et al. (2020) proposed the sequential fusion method PointPainting by projecting a LIDAR point cloud to the output of an image-only semantic segmentation network and attaching a category score to each point. The additional point cloud can then be provided to any LIDAR-only method. Paigwar et al. (2021) proposed Frustum-PointPillars. This method uses a 2D target detection field to reduce the search space.
in 3D space and then performs target localization based on a columnar feature encoding network for the simplified point cloud. Moreover, the Frustum-PointPillars algorithm can effectively mask the data beyond the target, improving the target localization capability. The onboard computers of unmanned vehicles are developing into embedded domain controllers, and the current mainstream fusion algorithms have significant advantages in the detection accuracy of targets, though. However, in actual applications, these algorithms have problems of complex models and high time complexity and cannot be applied to embedded devices in real-time.

In this paper, we propose the MDY algorithm, which uses 80-line LIDAR as the main and 16-line LIDAR is used to supplement point cloud information at blind spots and incorporate it into the image data, and the improved YOLOv3 algorithm for pedestrian detection and uses the point cloud data of LIDAR for pedestrian target ranging to obtain richer pedestrian information further. The main contributions of this paper are as follows:

(1) Our vision detector uses the YOLOv3 algorithm, which has been improved in terms of optimizing anchor frame, adding small target detection branches and TensorRT acceleration, as a way to improve detection accuracy and real-time performance.

(2) For the poor manual calibration accuracy of multisensor external parameters before fusion, an automatic labelling algorithm based on plane fitting of the LIDAR point cloud calibration plate is used, which greatly improves the calibration accuracy.

(3) We use the minimum obstacle bounding box construction algorithm based on the bird’s eye projection method to solve the problem of unstable wrapping of the obstacle bounding box by LIDAR.

(4) Our proposed MDY algorithm performs better on the KITTI dataset. It achieves effective accuracy and real-time performance in the actual application of the Jetson TX2, an on-board computer with low computing power.

2. Related work

2.1. Target detection algorithms

With the rapid development of deep learning, big data and hardware technology, deep learning techniques have been developed rapidly, and scholars have achieved remarkable results in computer vision. Target detection algorithms can be divided into two categories. The first category is a two-stage target detection algorithm based on candidate regions, which first generates a series of target candidate frames as samples. It then uses convolutional networks to classify samples of target candidates, with typical representatives such as Fast R-CNN (He et al., 2015) and Mask-RCNN (He et al., 2017). At present, two-stage target detection algorithms have the problems of extensive computation, complex processes and slow detection speed in the extraction of candidate regions. Although they have high accuracy, they cannot meet real-time demand. The second category is the regression-based single-stage target detection algorithm, which mainly uses the end-to-end idea of using the whole image to regress to predict the class and location of the target object, such as the SSD (Liu et al., 2016) and YOLO (Redmon et al., 2016) series algorithms. Among them, the YOLO algorithm can only input the same resolution as the training image and detects only one for a single frame with multiple objects leading to poor localization accuracy. Based on this, Redmon and Farhadi (2017) improved and optimized it and designed the YOLOv2 network to improve the detection performance of the algorithm. Then in 2018, Redmon and Farhadi (2018) introduced FPN, designed YOLOv3, and used the Darknet-53 network for feature extraction to improve its detection performance. YOLOv4 (Bochkovskiy et al., 2020) used the CSPResNext50 network, which has improved accuracy and speed but also has higher requirements on the computer hardware.

Based on the above considerations, this paper develops a study of the YOLOv3 algorithm using Darknet53 as a backbone network, extending the use of ResNet jump layer connections. Moreover, the two-step convolutional kernel is used instead of the stacked layers to prevent the negative gradients due to stacking and downsampling. And batch normalization (BN) is executed after each convolutional layer to prevent gradient disappearance and overfitting by the LeakyRelu activation function. In addition, YOLOv3 is acquired at three scales to prevent the loss of fine-grained features due to downsampling. During training, the network is trained and tuned using high-resolution images to improve target detection accuracy further. We improved pedestrian detection in actual applications, improving the algorithm accuracy and real-time performance.

2.2. Multisensor data fusion algorithm

For multi-sensor systems, the data from different types of sensors are different, so the essential requirement of data fusion algorithms is to parallelize the processing of this data to obtain better robustness. Currently, there are two main types of fusion algorithms commonly used: model-based approaches (Muresan et al., 2020;
Muresan & Nedevschi, 2019, September; Plangi et al., 2018) and data-driven approaches (Chen et al., 2017; Jason et al., 2018; Nie et al., 2021). Model-based methods such as probabilistic data association (PDA), multi-hypothesis trackers (MHT) and Unscented Kalman Filter (UKF) have become typical methods for data association and target localization. Muresan et al. (2020) proposed four approaches to address autonomous vehicle stabilization and verification tasks. The first raw data association method is proposed for tracking LiDAR targets, which combines multiple appearance and motion features to leverage the available information from road targets. The second approach proposes a tri-modal camera-based data association algorithm for sensor fusion target measurement correspondence, enriching sensor data by adding semantic information. The third approach proposes a model- and neural network-based data fusion method for multiple types of sensors with complementary data, using an Unscented Kalman Filter and a single-layer perceptron to fuse the information retrieved from five complementary sensors. On the other hand, the fourth approach uses fuzzy logic techniques combined with semantic segmentation of images to solve the problem of confirming the position of 3D objects, and the algorithm has better real-time performance. As the environment’s complexity increases, making the model-based approach suffers from factors such as large uncertainty of the model and an overly complex model. Thus, a data-driven approach is derived. The data-driven approach does not rely on a priori information and empirical knowledge and starts entirely from data, which is a top-down processing approach. Nie et al. (2021) proposed a data-driven multi-modal fusion framework (IMF-DNN) with strong robustness, generalization, and flexibility for target detection and end-to-end tasks.

Sensor fusion methods are divided into three main categories: pixel-level fusion, feature-level fusion and decision level fusion. Pixel-level fusion processes the received sensor data directly, which can fully utilize the original data but cannot guarantee the real-time performance of the system. Feature-level fusion performs feature extraction after obtaining data from each type of sensor, which can effectively fuse data from each type of sensor, and the system has better fault tolerance. However, it will discard part of the original data. Decision-level fusion is based on the decision or classification of each sensor independently, the recognition results of multiple sensors based on some rules to make the optimal solution of the system. The advantage of this kind of decision model is an excellent real-time and adaptive anti-interference ability which is primarily used in engineering practice. However, the decision performance greatly depends on the decision-making ability of a single system.

This paper synthesizes the advantages and disadvantages of different fusion methods of multi-sensors. From the perspective of actual applications, the MDY algorithm proposed in this paper uses a data-driven and model-based approach to fusion under a hybrid framework of pixel-level and decision-level, which ensures both the detection accuracy and real-time performance after fusion.

3. MDY algorithm

We adopt the decision-level and pixel-level hybrid fusion method to minimize data loss and reduce computational complexity. The proposed fusion framework is shown in Figure 1.

The detection framework is divided into three modules: the data matching module, the visual detection module, and the fusion and detection module.

In the data matching module, the three sensors are aligned temporally and spatially to obtain consistent results for the target. The visual detection module uses the decision level fusion structure to align the LiDAR point cloud data and the image pixel information. Then, the detection results are fused at the decision level. The specific steps are as follows:

(1) The improved YOLOv3 algorithm is used to extract features from the camera sensor. The decision’s category label and the target detection frame’s pixel position are obtained by feature matching.

(2) The point cloud data are preprocessed by ground point segmentation. Subsequently, the non-ground points are clustered to obtain the 3D detection frame of the target.

(3) The information of point cloud is transformed by a rotation translation matrix obtained by data matching and projected into the image’s pixel coordinate system. For targets detected by camera and LiDAR sensors, the decision level’s preliminary fusion is completed using the position relationship of the two detection boxes after transformation. For targets not detected by the camera but detected by LiDAR sensors, the shape of the pedestrian target and the characteristics of the pedestrian point cloud aggregation are used to judge, and the output category is pedestrian for the clustered targets that meet the conditions.

(4) The point pair of the point cloud and pixel data is fused at the pixel level, expanding the channel of the LiDAR data. The RGB information from the image is added to the point cloud data, and the category information from the image detection and the LiDAR detection frames are fused. Finally, the fusion and
Figure 1. Schematic diagram of the fusion framework-MDY algorithm. The 80-line LIDAR is the main sensor, and the 16-line LIDAR provides supplemental data for determining the target location from the building point cloud data. The image data are used for target recognition. The fusion algorithm combines the data from the three sensors to obtain stable target detection results.

3.1. Data matching

In the data matching module, the temporal and spatial data of the two LIDAR sensors are matched with the camera information to ensure consistency of the target information obtained from the different sensors.

1. Time matching. The sampling frequency differs for the different types of sensor data; thus, it is necessary to match the data acquisition time of different sensors. Therefore, the Robot Operating System (ROS) time synchronization method is used to find the image data frame closest to the LIDAR data acquisition time using a nearest neighbour method.

2. Spatial matching. The sensors are located in different positions of the car body and have different coordinate systems. It is necessary to transform the coordinate system of the three sensors into a unified coordinate system to obtain a consistent description of the target by the camera and LIDAR data. The coordinate system conversion principle is shown in Figure 2.

Because the LIDAR sensors and the camera are rigidly connected to the experimental vehicle, we only consider
the transformation relationship between the LIDAR sensor and the camera coordinate systems. The LIDAR coordinate system is the world coordinate system. The pixel coordinate system can be transformed into the world coordinate system using the rotation matrix \( R \) and the translation matrix \( T \). The transformation relationship between the world coordinate system and the camera coordinate system is described by equation (1):

\[
\begin{bmatrix}
    X_c \\
    Y_c \\
    Z_c \\
    1
\end{bmatrix} =
\begin{bmatrix}
    R & T \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    X_w \\
    Y_w \\
    Z_w \\
    1
\end{bmatrix}
\]

Based on the transformation relationship of the above coordinate system, the transformation relationship between the pixel coordinate system and the world coordinate system is described by equation (2):

\[
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix} =
\begin{bmatrix}
    f_u & 0 & C_u \\
    0 & f_v & C_v \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    X_w \\
    Y_w \\
    Z_w \\
    1
\end{bmatrix}
\]

Where \( f_u \) and \( f_v \) are the horizontal and longitudinal focal lengths of the camera, respectively, and \( C_u \) and \( C_v \) are the optical centers.

Obtaining the external parameters of the camera and the LIDAR units requires calculating the rotation and translation matrices by matching the feature points. A chessboard is used to match the point cloud feature points to the image feature points for different view angles. Manual calibration often has the characteristics of low accuracy, large error and so on. We use a multi-sensor automatic calibration method to improve the accuracy of the calibration.

A rectangular region of interest (ROI) is defined on the calibration board, and the region of the chessboard is intercepted by dynamically adjusting the ROI in the point cloud data. Next, the coordinates of the highest point in the point cloud data are obtained because the length and width of the chessboard are known. The point cloud coordinates of the lowest point of the calibration plate are obtained using the diagonal length. The point cloud can be intercepted according to the values of the highest and lowest point on the Z-axis to filter out the ground points in the point cloud data. Next, the point cloud plane of the chessboard calibration board is fitted, and the average vector of the plane and the corners of the calibration board are obtained. We use the least-squares method to fit the point cloud plane in the chessboard plane because the LIDAR points separated by plane fitting are not all located on the same plane. The fitting equation is defined in equation (3):

\[
ax + by + cz = d
\]

where \( a, b, c \) represent the normal vectors of the plane. The vectors satisfy the following relationship:

\[
a^2 + b^2 + c^2 = 1
\]

when \( d \) is the distance from the coordinate origin to the chessboard plane. The value is calculated using the least-squares method. The distance between the point and the plane is calculated for all points in the chessboard plane \( x_i, y_i, z_i \) according to the plane fitting equation. This distance is \( d_i \). A point belongs to the chessboard plane if \( |d_i - d| < \Delta d \), where \( \Delta d \) is the threshold. The segmented point cloud is then projected onto the fitted plane.

The RANSAC (random sample consensus) algorithm is used for fitting the straight lines to determine the corners of the chessboard’s four-line segments according to each side’s endpoints. The coordinates of the board’s four corners are obtained by correlating the interior points using the Point Cloud Library (PCL), and the central coordinate of the board can be determined. The blue arrow in Figure 3(a) is the average vector of the calibration plate.

The image feature points are extracted using the calibration board. The corners of the calibration board are determined using the findChessboardCorners function in the OpenCV function library. The corner coordinates in the image coordinate system are recorded. The extracted image features are shown in Figure 3(b). Since the actual size of the grids of the calibration board is known, the corner coordinates and the central point of the chessboard can be calculated according to the width and height of the chessboard.

According to the pixel coordinates of the image feature points and the three-dimensional coordinates of the LIDAR data, we obtain the transformation matrices of the two coordinate systems using the Perspective-n-Point (PnP) algorithm (Chen & Wang, 2018). Then, the external parameters are obtained. The rotation (Euler angle) and translation between the LIDAR and camera coordinate systems are obtained. An optimization function (Verma et al., 2019) is used to align the data from the three sensors and detect the target. The final calibration effect is shown in Figure 4.

### 3.2. Visual detection

Considering the need for real-time accurate sensing requirements of pedestrian targets for unmanned vehicles, we use the improved YOLOv3 algorithm for pedestrian target recognition. For the shortcomings in the application of this scenario, we made some improvements: optimizing anchor frames, adding small target
Figure 3. (a) shows the coordinate values of the four corners of the calibration plate and the schematic diagram of the checkerboard calibration plate’s central coordinates, vertices and central points. The figure’s blue arrow is the calibration plate’s average vector. (b) shows the figure of the image extraction feature.

Figure 4. (a) shows the final calibration effect of 16-line LIDAR. (b) shows the final calibration effect of 80-line LIDAR.

detection branches. In addition, the TensorRT (Zhou et al., 2020) acceleration technique is implemented to accelerate the algorithm, significantly improving the running speed.

3.2.1. Improved YOLOv3 algorithm

The anchor frame in the YOLOv3 algorithm is obtained from a K-means clustering algorithm to determine the initial position of the boundary box. The random selection of the K value may not necessarily provide the globally optimal result. Thus, the optimal anchor frame of the data set is obtained using the K-Means++ algorithm (Arthur & Vassilvitskii, 2006).

We add a small target detection branch to the original detection branch to improve the accuracy of small target detection; it contains 12 anchor frame sizes. The clustering accuracy improves from 82.53 to 88.68%. The anchor frame parameters obtained by the K-Means++ algorithm is listed in Table 1.

Different sizes of the same class of objects all may cause miss and detection errors in different scenarios. In the YOLOv3 target detection algorithm, the backbone network uses Darknet-53 and feature splicing to fuse the data obtained from three scales. The detection performance is better for large targets. However, since the convolution layer convolves the original image many times, shallow information is lost, resulting in omission errors of small target information. Therefore, we add a scale to extract the shallow information to improve detection accuracy. The modified network is shown in Figure 5.

3.2.2. YOLOv3 model acceleration

Due to the transition of unmanned vehicle computers to embedded devices with lower power consumption and miniaturization, most current target detection algorithms with high accuracy are based on neural networks. However, the computing power of the Jetson TX2 is only 1.33
Figure 5. Modified network. The black box part in the figure is the improvement point, adding a scale for shallow information, extracting shallow information and reducing the error detection rate of small targets.

Tera Operations Per Second (TOPS), which is insufficient for real-time detection using deep learning algorithms. In our paper, TensorRT acceleration technology is used to improve the learning speed of the model on Jetson TX2. According to the Darknet-53 network structure, the network is modified and optimized based on the following three aspects:

1. The useless output layer is eliminated to reduce the computation amount.

2. In the Darknet-53 network, to improve the model's generalization ability, prevent the gradient from disappearing in the model training process, and accelerate the convergence speed of the model, and adds a BN after each convolution layer. In the model training phase, each sample in the batch is normalized by calculating the mean $\mu$ and variance $\sigma$ of each batch data. The calculation method is shown in the equation (5), and the mean $\mu$ and variance $\sigma$ are calculated as shown in the equation (6).

$$\tilde{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$  \hspace{1cm} (5)$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \hspace{1cm} \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \hspace{1cm} (6)$$

Where $N$ is the size of the batch data, $x_i$ is each sample in the batch sample. Generally, the convolution output of the front layer of the BN layer is $(N, C, W, H)$, where $C$ represents the number of channels output, and $W, H$ represents the width and height of the output characteristic graph. At the same time, a scaling factor $\gamma$ and a shift factor $\beta$ are often added to the calculation of the BN layer. The input result of the sample after passing through the BN layer is shown in equation (7).

$$\tilde{x} = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \hspace{1cm} (7)$$

When the model is in the training stage, the convergence speed of the model can be accelerated. The generalization ability of the model can be improved by adding the normalization layer, but in the reasoning stage of the model, that is, when testing the model, the BN layer should switch from the training state to the test state, and the approximate mean and variance in the model training are adopted. However, the data has to be transferred twice in the convolution layer and BN layer, which will affect the detection speed of the model. The forward inference of the model is accelerated by merging the parameters in the BN layer and the convolution layer. The merging of the parameters of the BN layer and the convolution layer is shown in equation (8).

$$x_{out} = \sum_{j=0}^{n} \left( x_i \ast \frac{\gamma \ast w_j}{\sqrt{\delta^2 + \varepsilon}} \right) - \frac{\gamma \ast \mu}{\sqrt{\delta^2 + \varepsilon}} + \beta \hspace{1cm} (8)$$

where $w_i$ represents the convolution weight, which is a very small number to prevent the denominator from being 0.
3. The input data for the concat layer in the Darknet-53 network is directly fed into the following operation instead of performing a separate calculation after the concat layer.

3.3. Fusion and detection

The fusion and detection module obtains stable target category information and spatial location data. First, it is necessary to clip the LIDAR and image data to the same detection area. The LIDAR data are preprocessed using a ground point segmentation algorithm because no ground points are required for target detection. Then, the pedestrian target is detected by the improved YOLOv3 algorithm, and the LIDAR clustering algorithm obtains the minimum 3D bounding box of the target. The rotation and translation matrix are used on the 3D bounding box of the LIDAR data, and the 3D point cloud data are transformed into 2D data in the image coordinate system. Decision level fusion is performed using the discriminant condition. Next, pixel-level fusion is conducted for the target to expand the LIDAR data channel and obtain the target detection results from the camera data. Finally, the fusion results are output to the decision-making and control module of the unmanned vehicle.

3.3.1. Clipping the LIDAR data

We use 80-line LIDAR as the main and 16-line LIDAR is used to supplement point cloud information at blind spots in our car as shown in Figure 10. LIDAR sensors have 360° range, because we consider the car’s frontal range, we only consider the frontal 180° range of the LIDAR sensors. Due to the different ranges of the LIDAR sensors and the camera, it is necessary to clip the data from the three sensors to a common area before data preprocessing (Jin et al., 2020).

In the actual operation process, the camera view is the base view, and the point cloud closer to the front of the car needs to be cropped to retain only the field of view common to the LIDAR sensors and the camera. The clipped common area is the intersection of the merge of the two LIDAR areas and the camera. Figure 6 shows the clipped result of the three sensors.

In the obstacle perception of unmanned vehicles, effective removal of ground points can significantly improve the accuracy of obstacle detection. Therefore, ground information can be segmented by ground point fitting in obstacle information extraction. Common ground segmentation algorithms are as follows: it includes horizontal calibration, average vector, grid height difference, absolute height, ground point segmentation algorithm based on ray, etc. In this paper, ground point segmentation is completed by using a pseudo-scanline filtering algorithm (He & Tian, 2009). The results are shown in Figure 7. After ground points are filtered, only non-ground points are retained in the whole point cloud data, which helps improve the accuracy of subsequent non-ground point clustering.

3.3.2. Determining the minimum bounding box

The main purpose of the boundary box is to outline the target. The general method is to create a boundary box with the target at the center point of the obstacle. Establishing a boundary box may be problematic. Therefore, we use the minimum bounding box algorithm based on the bird’s eye view projection to outline the target. This method greatly improves the construction stability of the bounding box. The algorithm finds the edges of a given polygon. If AB represents an edge, the algorithm projects the points of the other polygons to the edge AB. Establishes a pair of intersection points with the maximum distance, representing one of the edges of the frame. Subsequently, the other side of the bounding box is obtained, and the smallest area is selected as the final bounding box. The schematic diagram of determining the minimum bounding box of the target is shown in Figure 8.

The process consists of the following steps: 1) Use the point cloud of the target as the input; 2) Calculate the 2D projection of the target, which is the polygon in the XY plane from a bird’s-eye view; 3) Calculate the points’ convex hull to obtain the target polygon’s vertices. 4) Sort the resulting vertices clockwise. 5) According to the sorted vertices, two points are on one edge, and the other points are projected to obtain rectangular regions of different sizes. 6) Select the rectangular box with the smallest border area as the minimum bounding box of the target.

3.3.3. Data fusion

We use a decision-level and pixel-level hybrid fusion approach to ensure high fusion and detection accuracy. This method exploits the advantages of different multi-sensor fusion methods. In the decision level fusion step, the target detection frame obtained from the images is fused with the 3D detection box obtained from the LIDAR point cloud data. The steps are as follows:

1. The improved YOLOv3 algorithm is used to detect the image’s input data and output the target’s detection frame information, including the category, confidence score, location, length, and width.
2. The preprocessed point cloud data are clustered, and the minimum bounding box of the target is obtained.
3. The point cloud data in the bounding box are transformed into pixels in the image pixel coordinate system using the rotation translation matrix obtained.
Figure 6. (a) shows the different ranges of the LIDAR sensors and the camera. The yellow area is the frontal 180° range of the 80-line LiDAR. The brown area is the frontal 180° range of the 16-line LiDAR. The green and blue area is the range of camera. The blue area is the clipped common area. (b) shows the sample picture of the clipped common area where the three sensors.

Figure 7. Effect picture of point cloud before and after pseudo-scanline filtering algorithm. (a) shows the raw point cloud data before pseudo-scanline filtering algorithm; (b) shows non-ground points after pseudo-scanline filtering algorithm.

Figure 8. Schematic diagram of determining the minimum bounding box of the targets.
by data matching. The transformation relationship is expressed by equation (9).

\[
\begin{align*}
    u_x &= \frac{\text{cloudpo int}_x * f_x}{\text{cloudpo int}_y + c_x} \\
    v_y &= \frac{\text{cloudpo int}_y * f_y}{\text{cloudpo int}_z + c_y}
\end{align*}
\] (9)

where \(u_x, v_y\) represents the coordinates in the pixel coordinate system of the image after the point cloud data have been transformed, \(\text{cloudpo}\) represents a point of the point cloud data in the bounding box, \(f_x, f_y\) represents the focal length of the camera, \(c_x, c_y\) represents the coordinates of the camera’s optical center.

(4) For targets detected by camera and LIDAR sensors, the pixel coordinates of the transformed point cloud data are associated with the target detection frame. The two-step discriminant method is adopted. First, the minimum and maximum pixel values of the coordinates in the point cloud data are obtained. The center coordinates of the target are obtained by combining the two data sets, and the distance between the center of the target and the target detection frame is calculated as follows:

\[
    \text{dis} = \sqrt{(x_c - x_l)^2 + (y_c - y_l)^2}
\] (10)

where \(x_c, y_c\) represents the center coordinates of the image detection frame, \(x_l, y_l\) represents the center coordinates of the point cloud data transformed into the pixel coordinate system. If \(|\text{dis}| < \epsilon\), the first step of data fusion is complete.

Many candidate frames are generated at the position of the same target, and they may overlap. The NMS algorithm finds the optimum target detection frame and prevents overlapping frames.

In the second step, the confidence scores of the image detection frames are sorted, and the frames with the highest confidence score are added to the output list. We calculate the intersection over union (IOU) with the other candidate boxes as follows:

\[
    \text{IOU} = 1 - \frac{S_c \cap S_l}{S_c \cup S_l}
\] (11)

where \(S_c\) represents the area of the image detection frame, \(S_l\) represents the area of the transformed point cloud data in the image coordinate system. If the IOU is less than the threshold \(\sigma\), the decision level fusion of the image detection and point cloud detection frames is completed. We repeat the above process until the output list is empty.

(5) Data fusion and information fusion are carried out on the image detection and point cloud detection frames after determining their relationship. A Red-Black Tree algorithm matches the pixel information and point cloud data, and the image information is combined with the point cloud data. The expanded point cloud data format is \((x, y, z, i, r, g, b)\), and the image information is fused with the point cloud clustering results. Finally, we can obtain the pedestrian information shown in Figure 9.

### 4. Experiment

#### 4.1. Experimental platform

A fourth-generation whirlwind smart car was used in the actual test on a university campus (Figure 10). The training and testing on the dataset are on Nvidia GTX 1080 GPU, cuda10.0, cudnn7.5, pytorch1.4 and python3.6. During the training of YOLOv3 and improved YOLOv3 algorithms, the size of an input image is 416 × 416, the iterations are 6500, the learning rate is 0.001, momentum is 0.9, and the Attenuation coefficient is 0.0005. The model was trained with different input resolutions using 100 iterations.

The specific information of the device installed on the whirlwind smart car are listed in Table 2.

| Device       | Installation location          |
|--------------|-------------------------------|
| LIDAR (80)   | Central axis of car roof      |
| LIDAR (16)   | Front part of the car         |
| camera       | Middle of the car windshield  |
| Jetson TX2   | Car trunk                     |

For targets not detected by the camera but detected by LIDAR sensors, the shape of the pedestrian target and the characteristics of the pedestrian point cloud aggregation are used to judge. The shape of the bounding box of the clustered pedestrian target is a vertical rectangular shape, and the size is much smaller than the bounding boxes of other obstacles such as cars in the scene, and unlike other clustered targets, the point cloud of pedestrians is mostly clustered in the Z-axis. With the least-square linear regression method to describe the main distribution direction of each cluster by a linear function. When the direction angle of a cluster is more than 30°, this cluster has a high possibility to be classified into person category. For the clustered targets that meet these above conditions the output category information is person.
Figure 9. The output effect after the MDY algorithm. (a) shows the effect that relying only on visual detectors; (b) shows the effect of MDY algorithm in this paper, fusing information from images and point cloud about the detected pedestrians.

Figure 10. Experimental platform which has a camera, 2 LIDAR sensors which one is 16-line, the other is 80-line and an on-board computer. The on-board computer is an embedded Jetson TX2 device which has 6-core CPU architecture, 256-core Pascal architecture GPU architecture, 8G of memory.

4.2. Evaluating indicator

The data set consisted of the Visual Object Classes (VOC) 2007 pedestrian target dataset (http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html/) and the KITTI (Geiger et al., 2012) dataset and the self-collected dataset during the experiment. The VOC dataset includes a training set (5011 samples) and test set (4952 samples), including 20 categories. We use the KITTI 3D object detection dataset for all our fusion experiments and evaluation. Detection results are evaluated based on easy, moderate and hard levels of difficulty. The KITTI dataset consists of samples with 3D point clouds, images and corresponding camera-LIDAR calibration data. The dataset is divided into 7481 training and 7518 testing samples.

The self-collected dataset included different scenes. All scenes were in the different areas of the north fourth ring campus of Beijing Union University and were acquired at different distances and with different light intensities. In order to increase the diversity of samples and prevent the occurrence of model overfitting in collected data sets, it is supplemented by pre-processing such as random clipping, rotation, translation, color change and other methods. Finally, the data set included 2832 pedestrian samples, including 2265 training data sets and 567 test samples. The pedestrians in the images had different postures, such as walking, sitting, and jumping, and included pedestrian targets of different ages and genders.

The Precision and Recall are typically used to evaluate the performance of the detection model. The Precision is the predicted ratio of positive detections in the total positive detection samples; it is calculated using equation (12).

\[
\text{Precision} = \frac{T_P}{T_P + F_P} \times 100\% \quad (12)
\]

\(T_P\) represents the number of positive samples predicted as a positive samples; \(F_P\) represents the number of negative samples predicted as a positive samples. The Recall is the ratio of the positive prediction in the total positive samples of the Ground truth; it is calculated using equation (13).

\[
\text{Recall} = \frac{T_P}{T_P + F_N} \times 100\% \quad (13)
\]

\(F_N\) represents the number of positive samples predicted as a negative samples. The average precision (AP) is often used to express the detection accuracy of a category; it is obtained using equations (12) and (13). The area under the Precision-Recall curve is the AP of a given category. We use the AP (IOU = 0.5) in image detection and AP_{3D} (3D IOU = 0.5) in fusion detection.

The detection accuracy is calculated as follows:

\[
D = \frac{T_D}{T_D + T_F} \times 100\% \quad (14)
\]

Where \(D\) is the detection accuracy of video frames; \(T_D\) represents the total number of video frames correctly detected; \(T_F\) represents the number of video frames missed and incorrectly detected.
4.3. Result analysis

A performance test of the improved YOLOv3 algorithm is carried out using the Jetson TX2 to determine the detection rates of the TensorRT and the accelerated TensorRT method. We used 100 frames in the test. The average results of the performance test are listed in Table 3.

As shown in Table 3, the run speed is 5.0 FPS before TensorRT acceleration. After TensorRT acceleration, the run speed is 16.3 FPS. The results indicate that the TensorRT acceleration has significantly improved the forward reasoning speed of the model, meeting the requirements for real-time detection by unmanned vehicles.

We compared the proposed improved YOLOv3 algorithm with three classical algorithms, Faster-RCNN, SSD and YOLOv3. The test results of the different algorithms are listed in Table 4.

For the same hyperparameter settings, the AP of the improved YOLOv3 algorithm is 76.2%, which is 6.4% higher than the original algorithm, and the run rate is 11.3 FPS faster than that of YOLOv3 on Jetson TX2.

The results of AP3D and FPS are listed in Table 5. The AP of the MDY algorithm is better on three levels, and the MDY algorithm achieves the best real-time performance. In this paper’s comprehensive consideration of detection accuracy and real-time performance, the MDY algorithm achieves the best performance.

Table 3. Comparison of algorithm performance before and after TensorRT on Jetson TX2.

| Technology | CNN/s | Detection/s | Run speed (TX2)/FPS |
|------------|-------|-------------|---------------------|
| Before TensorRT acceleration | 0.150 | 0.050 | 5.0 |
| After TensorRT acceleration | 0.041 | 0.0205 | 16.3 |

Table 4. The AP test result on the VOC2007 pedestrian target dataset and the self-collected dataset and run speed of different algorithms.

| Algorithm       | AP/% | Run speed (TX2)/ FPS |
|-----------------|------|----------------------|
| Faster-RCNN     | 62.6 | 1.1                  |
| SSD             | 53.0 | 2.3                  |
| YOLOv3          | 69.8 | 5.0                  |
| (Ours)Improved YOLOv3 | 76.2 | 16.3 |

Table 5. The result of AP3D on the KITTI test set and run speed of different fusion algorithms.

| Pedestrians_AP3D/% | Easy | Moderate | Hard | Run speed (GPU)/ FPS |
|--------------------|------|----------|------|----------------------|
| AWOD-FPN           | 50.80% | 42.81% | 40.88% | 9.9                  |
| MLOD               | 47.58% | 37.47% | 35.07% | 10.1                 |
| F-ConvNet          | 52.16% | 43.38% | 38.80% | 2.1                  |
| Painted PointRCNN  | 50.32% | 40.97% | 37.87% | 1.9                  |
| F-PointPillars     | 51.22% | 42.89% | 39.28% | 14.2                 |
| Ours (MDY)         | 52.03% | 42.95% | 39.01% | 26.1                 |

Table 6. The test results of 1000 frames in the north fourth ring campus of Beijing Union University by our fourth-generation whirlwind smart car.

| Algorithm     | TD | TF | D/% | Run speed (TX2)/ FPS |
|---------------|----|----|-----|----------------------|
| YOLOv3        | 724 | 276 | 72.4 | 5.0                  |
| (Ours)Improved YOLOv3 | 840 | 160 | 84.0 | 16.3                 |
| Ours (MDY)    | 908 | 92  | 90.8 | 15.9                 |

Figure 11. Results of pedestrians detecting in the different distances on Jetson TX2. (a)(1) shows the results of the YOLOv3 algorithm at a near distance; (a)(2) shows the results of the improved YOLOv3 algorithm at a near distance; (a)(3) shows the results of the YOLOv3 algorithm at a far distance; (a)(4) shows the results of the improved YOLOv3 algorithm at a far distance. (b)(5) shows the results of the MDY algorithm at a far distance; (b)(6) shows the results of the MDY algorithm at influence of light.

Other fusion algorithms do not achieve real-time well on GPU, so they cannot run on Jetson TX2, which has only 1.33 TOPS in real-time. To further verify the feasibility of the MDY algorithms, we test 1000 frames with our fourth-generation whirlwind smart car, which on-board computer is Jetson TX2 at the North 4th Ring Campus of Beijing Union University. The results are shown in Table 6 and Figure 11.
Table 6 shows that the detection accuracy of pedestrians is 72.4% for the YOLOv3 algorithm and 84.0% for the improved YOLOv3 algorithm (an increase of 11.6%), demonstrating the superior performance of the improved algorithm. Figure 11(a) shows the actual effect before and after improved YOLOv3 algorithms. However, the distance and light conditions affect target detection based on imagery, leading to missed targets. The incorporation of LiDAR data can avoid these problems effectively. The MDY algorithm has higher accuracy than using only image data with almost no loss of speed shown in Table 6. Figure 11(b) shows that the MDY algorithm provides better results than the image data alone. Excellent performance of the MDY algorithm was achieved in our actual application of this paper.

5. Conclusion
To address the shortcomings of a single sensor and the low computational power of embedded on-board processors, the MDY algorithm is proposed in this paper, which combines accuracy and real-time performance in our actual applications. In order to improve the detection accuracy of the vision detector, this paper improves the YOLOv3 algorithm. For the limited computational power of embedded devices, this paper optimizes the YOLOv3 algorithm and accelerates it with TensorRT technology to meet the real-time requirements of unmanned vehicle detection in this paper. In order to balance the real-time performance and accuracy of the fusion algorithm, a hybrid fusion framework is used to complete the data fusion between multiple sensors. After actual testing, the MDY algorithm achieves 90.8% detection accuracy based on ensuring real-time performance. Although the MDY algorithm has been able to perform well in pedestrian detection, the information of LiDAR is not fully utilized because the camera view cannot cover more distant information. Future research will further consider the fusion of long and short focal length camera view data to enhance the camera field.

Acknowledgement
This work was supported by the National Key R&D Program of China under Grant No. 2021YFC3001300, the National Natural Science Foundation of China under Grant No. 61931012, and the project supported by Beijing Natural Science Foundation under Grant No. 4222025 and supported by the Academic Research Projects of Beijing Union University (No. ZK10202208).

Disclosure statement
No potential conflict of interest was reported by the author(s).

Funding
This work was supported by the National Key Research and Development Program of China: [Grant Number 2021YFC3001300]; the National Natural Science Foundation of China: [Grant Number 61931012]; Beijing Natural Science Foundation: [Grant Number 4222025]; the Academic Research Projects of Beijing Union University â–ÁUGrant Number ZK10202208â–Á.

References
Arthur, D., & Vassilvitskii, S. (2006). k-means++. The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, SIAM, 1027–1035.

Bochkovskiy, A., Wang, C.Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Chen, P., & Wang, C. (2018). Iepnp: An iterative camera pose estimation algorithm based on EPnP. Acta Optica Sinica, 38(4), 0411001–144. https://doi.org/10.3788/AOS201838.0411001

Chen, X., Ma, H., Wan, J., Li, B., Xia, T., & Cao, D. (2017, July, 21–26). Multi-view 3D object detection network for autonomous driving. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 6526–6534. https://doi.org/10.1109/cvpr.2017.691

Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., & Cao, D. (2020). Deep learning for image and point cloud fusion in autonomous driving: A review. IEEE Transactions on Intelligent Transportation Systems, 722–739. https://doi.org/10.1109/tits.2020.3023541

Deng, J., & Czarnecki, K. (2019, October). MLDO: A multi-view 3D object detection based on robust feature fusion method. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC). 279–284. IEEE. https://doi.org/10.1109/itsc.2019.8917126

Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? The KITTI vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, 3354–3361. IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN//Proceedings of the IEEE International Conference on Computer Vision. 2961–2969.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916. https://doi.org/10.1109/tpami.2015.2389824

He, Z., & Tian, Y. (2009). Filtering algorithm for non-ground point from airborne laser scanner data. Journal of Geodesy and Geodynamics, 29(004)(4), 97–101.

Huang, K., Shi, B., Li, X., Li, X., Huang, S., & Li, Y. (2022). Multimodal sensor fusion for auto driving perception: A survey. arXiv preprint arXiv:2202.02703.

Jason, K., Melissa, M. a., Jungwook, L., Ali, H., & Steven L. W. (2018, October, 1–5). Joint 3D proposal generation and object detection from view aggregation. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–8. https://doi.org/10.1109/iros.2018.8594049

Jin, Y. H., Hwang, I. T., & Lee, W. H. (2020). A mobile augmented reality system for the real-time visualization of pipes...
in point cloud data with a depth sensor. *Electronics*, 9(5), 836. https://doi.org/10.3390/electronics9050836

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multiBox detector. European Conference on Computer Vision. Singapore: Springer International Publishing, 21–37.

Muresan, M. P., Giosan, I., & Nedevschi, S. (2020). Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation. *Sensors*, 20(4), 1110. https://doi.org/10.3390/s20041110

Muresan, M. P., & Nedevschi, S. (2019, September). Multi-object tracking of 3D cuboids using aggregated features. *IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 11–18. https://doi.org/10.1109/ICCP48234.2019.8959552

Nie, J., Yan, J., Yin, H., Ren, L., & Meng, Q. (2021, June). A multimodality fusion deep neural network and safety test strategy for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(2), 310–322. https://doi.org/10.1109/TIV.2020.3027319

Paigwar, A., Sierra-Gonzalez, D., Erkent, O., & Laugier, C. (2021). Frustum-pointpillars: A multi-stage approach for 3D object detection using RGB camera and LIDAR. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2926–2933. https://doi.org/10.1109/ICCVW54120.2021.00327

Plangi, S., Hadachi, A., Lind, A., & Bensrhair, A. (2018, December). Real-Time vehicles tracking based on mobile multi-sensor fusion. *IEEE Sensors Journal*, 18(24), 10077–10084. https://doi.org/10.1109/JSEN.2018.2873050

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788. https://doi.org/10.1109/cvpr.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6517–6525. https://doi.org/10.1109/cvpr.2017.690

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv: 1804.02767.

The PASCAL Visual Object Classes Challenge. (2007). (VOC2007). Results. [online] http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html (accessed December 2021).

Verma, S., Berrio, J. S., Worrall, S., & Nebot, E. (2019, October, 27–30). Automatic extrinsic calibration between a camera and a 3D LIDAR using 3D point and plane correspondences. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 3906–3912. https://doi.org/10.1109/ITSC.2019.8917108

Vora, S., Lang, A. H., Helou, B., & Beijbom, O. (2020). Pointpainting: Sequential fusion for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4603–4611. https://doi.org/10.1109/cvpr42600.2020.00466

Wang, Z., & Jia, K. (2019, November). Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1742–1749. IEEE.

Zhou, L. J., Liu, Y., Bai, L., Liu, F., & Wang, Y. W. (2020). Using TensorRT for deep learning and inference applications. *Journal of Applied Optics*, 41(2), 337–341. https://doi.org/10.5768/JAO2041.02020007