IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 30, 2021, accepted October 13, 2021, date of publication October 15, 2021, date of current version October 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120720

Multi-Modal Neural Feature Fusion for Automatic Driving Through Perception-Aware Path Planning

ZHENYU LI, (Student Member, IEEE), AIGUO ZHOU, JIAKUN PU, AND JIANGYANG YU
School of Mechanical Engineering, Tongji University, Shanghai 201804, China

Corresponding author: Zhenyu Li (zhenyu.li@tongji.edu.cn)

ABSTRACT Path planning is a significant and challenging task in the domain of automatic driving. Many applications, such as autonomous driving, robotic navigation and aircraft object tracking in complex and changing urban road scenes, need accurate and robust path planning by detecting obstacles in the forward direction. The traditional methods only rely on the path search method without considering the environmental factors, the vehicle path planning method cannot deal with the complex and changeable environment. To deal with above problems, we propose a perception-aware based multi-modal feature fusion approach that combines visual-inertial odometer (VIO) poses and semantic obstacles in the forward scene of vehicles to plan driving paths. The proposed method takes environment awareness as the guide and combines path search algorithm to realize path optimization task in complex environment. The proposed approach first uses a long short memory network (LSTM) to build a VIO that fuses visual and inertial data for pose estimation. To detect obstacles, the proposed method uses a segmentation model with a lightweight structure to extract semantic 3D landmarks. Finally, a path search strategy combining an A* algorithm and visual information is proposed to plan driving paths for intelligent vehicles. We estimate the proposed path planning method on assimilated scenes and public datasets (KITTI and Cityscapes) by using a micro controller (Jetson Xavier NX) installed on a small vehicle. We also show comparable results with path planning that only uses the greedy algorithm or heuristic algorithm without using visual information and show that our method is adequate in coping with different complex scenes.

INDEX TERMS Automatic driving, path planning, pose estimation, scene classification, VIO, obstacle detection, semantic landmarks.

I. INTRODUCTION

Visual perception is an essential part of automobile intelligence. As the autonomous driving industry continues to advance, the demands of perception-related technologies are also growing continuously, which undoubtedly pushes forward the rapid development of technologies. In recent years, LIDAR is widely used in visual perception [1]–[4] for automatic driving.

Although LIDAR has an advantage in accuracy and robustness, the price is expensive, which undoubtedly increases the development and manufacturing costs of intelligent vehicles. Therefore, many researchers have begun to focus on visual perception methods based on low-cost cameras, all of whom use only a low-cost camera or cameras for performing automatic driving tasks [5]–[8]. The related applications mainly include localization and navigation, lane detection, traffic light identification, pedestrian detection and building recognition. In general, visual perception is the foundation of path planning, and the key to path planning is the ability to help moving vehicles achieve pose adjustment to avoid detected obstacles in front of the vehicle. Visual perception not only includes pose estimation but also includes obstacle detection. In the early years, researchers obtained pose estimation results of robots by a single camera or IMU, which shows many advantages over radar, such as being low in cost, light in weight, and easy to install [9]–[12]. There are some drawbacks, however, including detection accuracy, distance, stability and being easily disturbed by the external environment. For example, when the light changes, the images captured by the camera are easily blurred. Additionally, the IMU is prone to producing a large drift error when the vehicle turns sharply. These drawbacks are not conducive to realizing automatic driving.

The associate editor coordinating the review of this manuscript and approving it for publication was Khin Wee Lai.

To address these problems, researchers have proposed alternative solutions, in which the most widely used solution is to fuse the camera with IMU to compensate for the disadvantage of only having a single sensor. The existing approaches mainly include VIFTrack [13], Maplab [14], Iceba [15], and PIVO [16], which almost always use fused data for some recognition tasks such as feature tracking, automatic navigation or pose estimation. By comparing these approaches with a single sensor approach, we can see that VIOs have better performance for visual recognition tasks. However, these methods can only cope with problems of localization and navigation and are unable to cope with the problem of path planning for automatic driving. It is challenging for intelligent vehicles to drive autonomously in complex urban street scenes. Intelligent vehicles need to have the ability of scene classification as well as localization and navigation. At the same time, vehicles need to detect categories of objects ahead of them in order to develop a strategy for obstacle avoidance based on a path planning algorithm. However, none of the existing methods address the problem of scene classification in terms of path planning in automatic driving. The proposed scene classification methods only focus on object recognition or local scene recognition rather than a path planning strategy [17]–[19].

To address these problems, a novel path planning method combining VIO and scene information for path planning is proposed in the paper. In summary, the main contributions of this paper are shown as follows:

(1) we propose a perception-aware path planning method that combines visual pose estimation, obstacle detection and path searching algorithm.

2) we propose to use a deep VIO model to output pose information.

3) we construct a lightweight semantic segmentation network for obstacles detection.

4) The proposed method is evaluated on a public urban street scene benchmark and a simulative environment based on a Jetson Xavier NX installed on a small vehicle using the standards of existing methods.

II. RELATED WORK

In this section, we discuss the contents closely related to our proposed method. The advanced nature and limitations of some methods are discussed, and the problems and advantages that we can solve are also discussed.

A. POSE ESTIMATION

In computer vision, estimating pose is a challenging task for robots, especially with a weak GPS signal or without a GPS environment. This is mainly reflected in the directional adjustment of vehicles when driving. The most popular approach to address pose estimation is to combine visual and inertial information, which has many advantages over only using a single sensor, like a camera or an IMU. As we know, the camera is prone to inaccuracy in extreme environments, for example, weak illumination, heavy rain or dense smog. The IMU also has a disadvantage in that it will cause accumulative drift error over time. Therefore, fusing visual and inertial data combines the complementary advantages of the two kinds of sensors in an intelligent vehicle navigating to a destination. For example, Zhu *et al.* [20] proposed an algorithm that fuses a pure event-based tracking algorithm with an IMU to provide 6-DoF camera pose tracking. Alatise and Hancke [21] proposed a low-cost and precise location method for mobile robots through combining a 6-DoF IMU (including three axes of the accelerometer and three axes of the gyroscope) with a camera. Because of motion blur and latency, visual data from capturing cameras are incomplete, which presents a challenge for pose capture. To cope with this, some bio-inspired vision methods have been proposed. For example, Mueggler, *et al.* [22] proposed using a continuous-time representation to perform VIO through an event camera that could deal with the high temporal resolution and asynchronous characteristics of the sensor. Rebecq, *et al.* [23] presented a novel, accurate and tightly coupled VIO pipeline, which exhibited outstanding performance in estimating the camera's self-motion under challenging conditions. To track non-corner shaped features, Bloesch, *et al.* [24] proposed using an iterated extended kalman filter (IEKF) to fuse tight inertial measurements with visual data from one or more cameras to form a robust VIO. With the rapid development of deep learning, some deep-VIOs based on deep neural networks show many advantages over these traditional methods. The deep VIOs have the advantages of self-extraction and self-coding. Furthermore, the extracted neural features are robust in extreme environments. In recent years, some deep VIOs, e.g., DeepFuse [25], DeepVIO [26], Vinet [27], and Self-VIO [28], [29], have been proposed for pose estimation and perform better than traditional methods. In our work, we use two kinds of deep learning models (CNN and LSTM) to encode visual and IMU frames. Then, we fuse the two kinds of data from a camera and an IMU within the VIO for pose estimation.

B. SCENE SEGMENTATION

Scene segmentation is a fundamental task of environmental modelling in automatic driving. In an urban scene, the ability to classify every object (vehicles, buildings, trees, and pedestrians) around the vehicles is necessary for safe driving. The key task for vehicles is to identify each obstacle that appears in the front of the vehicle by pixel-level segmentation. Pixels existing in the high-resolution image are captured by a camera mounted on a vehicle, and need to be classified as a set of semantic tags. Differently from other scenes, the scale variation of objects in the automatic driving scene is large, which brings great challenges to high-level feature representation. Therefore, encoding the multi-scale information correctly is imperative. Yang, *et al.* [30] proposed introducing "atrous" convolution to generate features with larger receptive fields without sacrificing spatial resolution. Luc, *et al.* [31] presented an autoregressive CNN architecture that learns to generate multiple frames iteratively. Due to the large difference between labeled training or source data in the real world, significantly decreased performance is observed which cannot be easily remedied. To overcome this problem, a novel UDA framework [32] based on an iterative self-training (ST) procedure was proposed, which is meant to address this difference through latent variable loss minimization. Most semantic segmentation methods rely heavily on pre-trained networks, which were originally used to classify images as a whole. Although the networks show excellent recognition performance, these methods perform poorly in terms of localization accuracy. To address this problem, Pohlen, *et al.* [33] presented a novel architecture that has strong localization and recognition abilities without pre-training. Because of the extraordinary computational complexity involved, many research works have proposed using lightweight architectures to decrease the computational complexity by reducing the number of network layers. For example, Orsic, *et al.* [34] presented an alternative approach which achieves a significantly better performance across a wide range of computing budgets. In our work, we also present a lightweight model with a bottleneck structure based on ResNet for semantic detection, which is end-to-end trainable.

C. PATH PLANNING

The task of scene perception is to help intelligent vehicles distinguish their surrounding environment. Additionally, the vehicle needs to have the ability to avoid obstacles in complex environments. In general, the shortest and most efficient path from the starting point of the design to the final destination, while avoiding obstacles during automatic driving, is the primary goal of most research. In recent years, studies have focused on the path planning of unmanned ground vehicles. The goal is to find the shortest, most stable, most economic and safest path under the conditions of collision avoidance, motion boundary and speed constraints in the presence of obstacles and water flow. For example, Ma, *et al.* [35] referred to this problem as a multi-objective nonlinear optimization problem with generous constraints and proposed to adopt an augmented multi-objective particle swarm optimization algorithm to find the solution. Broumi, *et al.* [36] applied the Dijkstra algorithm to solve the shortest path problem. However, while the traditional Dijkstra algorithm can find the shortest path, it skips over other paths with the same distance and cannot deal with the isometric shortest problem. To address this problem, an improved Dijkstra algorithm is proposed to solve the problem of path planning in a rectangular environment, and it can determine all equidistant shortest paths [37]. The problem of path planning can also be treated as an engineering optimization problem, in which the shortest collision free path is taken as the criterion to optimize the driving path [38]–[40]. With the rapid development of machine learning, learning-based path planning shows many advantages. Wang, *et al.* [41] propose a learning-based technique to address the problem of replanning when the robot encounters a conflict. Wen, *et al.* [42] proposed using a fully convolutional residual network to recognize the obstacles and obtain depth images. The avoidance obstacle path is planned by the "dueling DQN" algorithm in the robot's navigation that exploits environmental spatial-temporal information. Zhou, *et al.* [43] proposed to use deep reinforcement learning algorithms for USV and USV formation path planning, which specifically focuses on safe obstacle avoidance in maritime scenes. In real-world scenarios, considering an ideal situation with a known environment, workable maps for real applications are big. To deal with above problems, Orozco-Rosas, *et al.* [44], [45] propose a hybrid path planning algorithm based on membrane pseudo-bacterial potential field, which is able to reduce the computational or time.

However, the above methods do not have scene information that can provide vehicles with a reference of actionable decisions. These methods have to rely on a greedy strategy to make the search path longer. Therefore, we propose a novel path planning method that combines prior visual information and the A* algorithm. Like Dijkstra, the A* algorithm is generally used to search for the shortest paths; also similarly to BFS, it can guide itself with a heuristic function. Because of its many advantages, applications based on the A* algorithm have been proposed to deal with path problems [46]–[48]. In this paper, we also use the A* algorithm to plan the vehicle's moving path by combining previous pose estimation and scene segmentation.

III. METHODOLOGY

In the paper, we propose a perception-aware path planning approach that combines multi-modal data and a path search algorithm. The proposed approach consists of four modules: feature encoding, pose estimation, obstacle detection and path planning, as shown in Fig. 1. The first module is mainly responsible for visual feature coding of the original data (visual and IMU data). The second module performs obstacle detection by using a lightweight segmentation network with an encoder-decoder. The third one performs pose estimation by a deep VIO that is constructed by using an LSTM. Finally, the last one performs path planning by combining scene perception and a path search algorithm for developing a safe obstacle avoidance strategy.

A. POSE ESTIMATION WITH A FUSION OF MULTI-MODAL NEURAL FEATURES

In this section, we encode an image and IMU frame by a lightweight CNN and LSTM and build a deep VIO to fuse two kinds of feature vectors that are used to estimate 6-DoF poses of the vehicle. After encoding, the image and IMU frame are translated into a visual feature vector and an inertial feature vector:

$$\boldsymbol{X_v} = \left[x_{v,t_1}, x_{v,t_2}, \ldots, x_{v,t_n}\right] \tag{1}$$

$$\boldsymbol{X_i} = \left[x_{i,t_1}, x_{i,t_2}, \ldots, x_{i,t_n}\right] \tag{2}$$

where $\boldsymbol{X_v}$ is the visual feature vector, and $\boldsymbol{X_i}$ is the inertial feature vector. x_{v,t_k} and x_{i,t_k} correspond to two kinds of modal frames that appear at the same time.

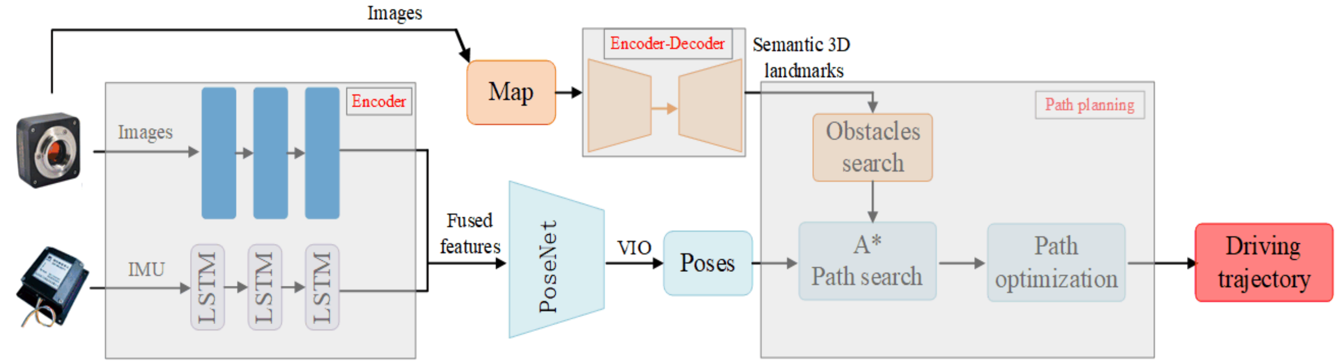


FIGURE 1. Overview of the proposed path planning framework. The whole process consists of four modules, namely, feature encoding, pose estimation, semantic segmentation and path planning. In the feature extraction module, an encoder based on CNN and an encoder based on LSTM are used for two kinds of data modal feature encoding. In the path planning module, the A* algorithm is considered to combine semantic landmarks for path searching and optimization.

Since the frequency of data gain is different and that of the IMU (100Hz ∼ 500Hz) is about ten times that of the camera (10Hz ∼ 50Hz) data gain. To cope with this problem, an LSTM is utilized to encode the inertial frame. Then, the visual and inertial data are fused together, which forms a new feature vector x_{f,t_k}:

$$\boldsymbol{X_f} = \left[x_{f,t_1}, x_{f,t_2}, \ldots, x_{f,t_n}\right], \quad x_{f,t_k} = x_{v,t_k} \oplus x_{i,t_k} \tag{3}$$

Furthermore, we introduce the LSTM to build a VIO, which is used for pose estimation. Given an image-IMU sequence, a mapping relation between the original frame and feature space is constructed by transforming the input sequences of images and IMU data to a space vector:

$$\boldsymbol{X_p} = \left\{x_{f,t_i} = x_{v,t_k} \oplus x_{i,t_k} \mid \left(x, y, z, q_x, q_y, q_z, q'\right)_{1:n} \in (\mathrm{R}^7)_{1:n}\right\} \tag{4}$$

In our work, we consider translation and rotation as motion on the manifolds and decompose the camera's motion into a rotation motion and a translation motion in the vector space. Then, the rotation motion and translation motion are translated into a rotation matrix and a translation matrix by a special Euclidean group $\boldsymbol{SE}(\mathbf{3})$:

$$\boldsymbol{SE}(\mathbf{3}) = \left\{T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} | R \in \boldsymbol{SO}\,(3)\,, t \in \mathbb{R}^3\right\} \tag{5}$$

Every Lie group $\boldsymbol{SE}(\mathbf{3})$ has a matching Lie algebra $se\,(3)$ that describes the local properties of Lie groups, represented as a vector ϕ defined in $\boldsymbol{SE}(\mathbf{3})$. We built an LSTM network to cope with the camera's motion on the manifold. The LSTM performs the mapping from the image data to the Lie algebra $se(3)$. Every moment of the camera's motion has a matching Euclidean group $\boldsymbol{SE}(\mathbf{3})$, which forms a trajectory flow by recording the time sequence. The record of the camera's motion is shown in Fig. 2. Next, the inertial vector is fed into three fully connected neural layers for pose regression. The fully connected neural layers regress 3D translations and 4D rotations as a quaternion and map the fused feature vector into a vehicle's pose vector x_p:

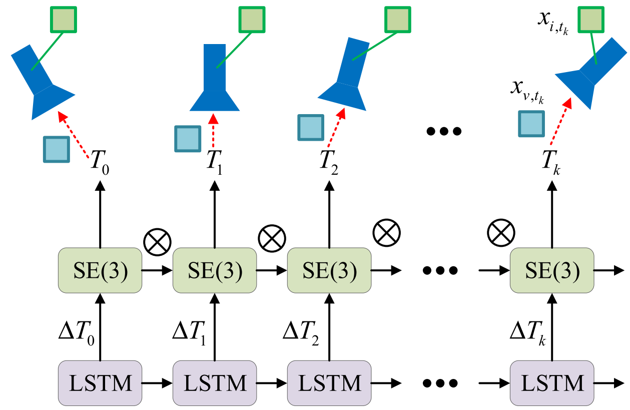


FIGURE 2. Illustration of the camera's motion on the manifold by using LSTM for recording sequences of frames, e.g., visual frame $\boldsymbol{x_{v,t_k}}$ and visual inertial $\boldsymbol{x_{i,t_k}}$.

$$x_p = \mathit{lstm}\left(x_{f,t}, x_{f,t-1}\right) \tag{6}$$

B. SEMANTIC 3D OBSTACLE DETECTION

Semantic segmentation is one of the key problems in computer vision. On the one hand, semantic segmentation is a high-level task that paves the way for an understanding of the scenes. The importance of scene understanding as a core computer vision problem lies in an increasing number pf applications that provide "nutrition" by inferencing knowledge from images. Some of these applications mainly include self-driving vehicles, human-computer interaction, virtual reality, etc. In recent years, with the popularity of deep learning, researchers have constructed some complex networks with a deeper and deeper structure to address semantic segmentation problems. The most common one is the convolutional neural network, which exceeds traditional methods in both accuracy and efficiency. On the other hand, semantic segmentation can predict pixel-level object categories. Generally, scene segmentation provides a comprehensive scene description

TABLE 1. The parameter set of The proposed network architecture. The size of output is 512 × 512.

Name	Type	Output size
Initial	-----	16×256×256
Bottleneck-1.0	Down-sampling	64×128×128
Bottleneck-1.x (x=2,3,4,5,6,7,8,9)	-----	64×128×128
Bottleneck-2.1	Down-sampling	128×64×64
Bottleneck-2.x (x=2,3,4,5,6,7,8,9)	dilated 2(2,2), asymmetric 5(2,3), dilated 4(2,4), dilated 2(2,6), asymmetric 5(2,7), dilated 4(2,8)	128×64×64
Bottleneck-3.x (x=2,3,4,5,6,7,8,9)	-----	128×64×64
Bottleneck-4.1	Up-sampling	64×128×128
Bottleneck-4.x (x=2,3,4,5,6,7,8,9)	-----	64×128×128
Bottleneck-5.1	Up-sampling	16×256×256
Bottleneck-4.x (x=2,3,4,5,6,7,8,9)	-----	16×256×256
Full-conv	-----	C×512×512

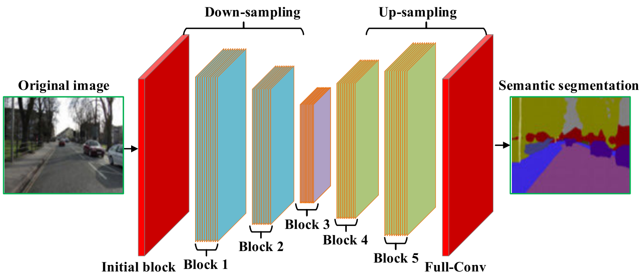


FIGURE 3. Overall architecture of the proposed asymmetric semantic segmentation network.

for vehicles, including object categories, location and shape. In our work, we present a lightweight asymmetry semantic segmentation model consisting of three parts: initial block, down-sample and up-sample, as shown in Fig. 3. Due to combining the advantage of ResNet, the proposed segmentation model has a strong ability to perform pixelwise classification. In addition, in order to reduce computing cost and improve computing efficiency in reference to E-net [47] we reconstruct this network using many bottleneck layers to reduce weights. The proposed network architecture is shown in Table 1. The semantic segmentation network consists of two parts: encoder (down-sampling) and decoder (down-sampling). The "Bottleneck" structure includes two filters with the size of 1×1, which is used for lower and raise feature dimensions. In the stage of down sampling, the output of feature dimension decreases with multiple ratios. In the stage of up sampling the output of feature dimension increases with multiple ratios. For the encoder, when given an image I that is first imported into a lightweight CNN, the encoder will output m intermediate features by using k convolutional filters $F_m = \left\{F_1^{w1}, F_2^{w2}, \ldots, F_k^{wk}\right\}$. The encoder will produce weight $W = \sum_{i=1}^{k} w_k$ and bias $B = \sum_{i=1}^{k} b_k$ through the following:

$$T_m = s\left(I * F_m^w + b_m\right), \quad m \in N \tag{7}$$

where s is the RRelu activation function, and b are the bias vectors for the k^{th} feature map. The decoder will also output m intermediate features by the previous down-sample T_m through the convolutional filters $F'_m = \left\{F_1^{w'_1}, F_2^{w'_2}, \ldots, F_k^{w'_k}\right\}$. During this process, the weight $W' = \sum_{i=1}^{k} w'_k$ and bias $B' = \sum_{i=1}^{k} b'_k$ will be produced by the following:

$$G_m = s\left(T_m * F_m^{w'} + b'_m\right), \quad m \in N \tag{8}$$

The parameters $\{W, B\}$ produced in the network are learned automatically through multiple iterations, which are used to achieve the optimal allocation to reduce the predicted loss. Therefore, according to the reconstructed function D_m, the last convolutional layer existing in each convolutional block will output n intermediate features by combining the previous up-sample G_m and using the convolutional filters $F''_m = \left\{F_1^{w''_1}, F_2^{w''_2}, \ldots, F_k^{w''_k}\right\}$. During this process, the weight $W'' = \sum_{i=1}^{k} w''_k$ and bias $B'' = \sum_{i=1}^{k} b''_k$ will be produced by the following:

$$D_m = s\left(G_m * F_m^{w''} + b''_m\right), \quad m \in N \tag{9}$$

Additionally, scene segmentation tries to produce labels for each pixel based on different categories appearing in a scene, as shown in Fig. 4. In our work, we select RRelu as the activation function and select softmax as the object classifier to detect all objects. According to the semantic features previously calculated, the classifier can detect every object appearing in the scene as follows:

$$B_m^* = softmax\left\{D_1, D_2, \ldots, D_m\right\}, \quad m \in N \tag{10}$$

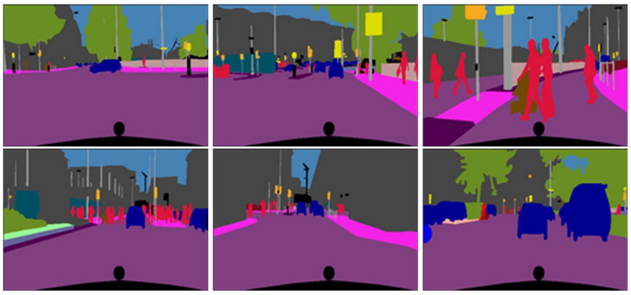


FIGURE 4. **Image segmentation label instances. The different color blocks represent the object categories, e.g., vehicles, trees, pedestrian, etc. The images are taken from the Cityscapes dataset.**

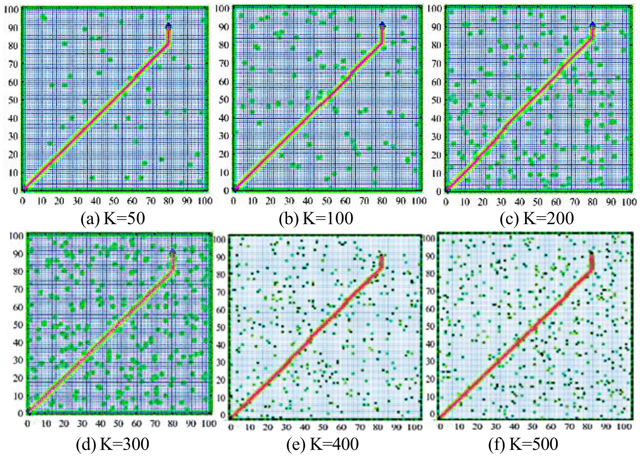
C. PATH PLANNING FOR AUTOMATIC DRIVING

Path planning is a key step towards automatic driving for traditional vehicles. Combining previous scene segmentation, intelligent vehicles can detect obstacles distributed around a vehicle by identifying semantic information. According to path planning, the intelligent vehicles can also create an obstacle avoidance strategy. Combining the path search algorithm, the vehicle can optimize the selected path to make the best driving path. Based on previous pose estimation, the intelligent vehicle can also avoid obstacles by adjusting the driving poses. In our work, we learned from heuristic search methods used in games, combining the A* algorithm and prior visual detection, to find a safe driving path. The main function of the path-finding strategy is to find the best path for obstacle avoidance and to minimize the cost (fuel, time, distance, equipment, and money). Compared with other algorithms, A* not only has high search accuracy but also fast search speed through combining previous visual information, which can meet the technical requirements of future self-driving vehicles.

The path planning consists of the starting point, the end point, all possible paths between the two points, the intermediate nodes (the intermediate state) and the search cost. In general, the path search uses a heuristic function, namely, the approximate cost from any node to the end point, to estimate the reward value quickly. The output is the optimal path from the start node to the end point, that is, the least costly. A good heuristic function will make this search operation as efficient as possible or search for as many paths as possible. Therefore, the cost estimate from the initial state can be calculated as follows:

$$f(n) = g(n) + h(n), \quad n \in N \tag{11}$$

where $g(n)$ is the actual cost from the initial state to state n in the state space, and $h(n)$ is the estimated cost of the best path from state n to the target state. The A* algorithm traverses all possible paths from the starting point, checks all possible extension points (adjacent points), and calculates $g(n) + h(n)$ for each point. Among all possible extension points, this algorithm will select the point with the smallest $f(n)$ as an extension automatically, that is, it calculates the $f(n)$ of all possible extension points at this point and adds these new extension points to the "Open List". By setting different numbers of obstacles in the space, the search time and nodes may be changed. Here, we randomly generate 500, 400, 300, 200, 100 and 50 obstacles to detect the search state with different numbers of obstacles, as shown in Table 2. The results are also shown in Fig. 5, which shows that the search time increases with the increase of obstacles under the unchanged search distance and changed search nodes. In addition, the increase of obstacles will raise the difficulty of path planning under the same starting point and target point conditions, which can also be shown from the continued increase in search time.



(a) K=50 (b) K=100 (c) K=200

(d) K=300 (e) K=400 (f) K=500

FIGURE 5. **Path search results with different numbers of obstacles. The red line represents the path result, the green box represents obstacles, and the blue rhombic box represents the start and end points of the path.**

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we discuss the establishment of the experiment, including the selection of datasets, the setting of experimental parameters and the layout of experimental scenes, and also discuss and analyze the experimental results.

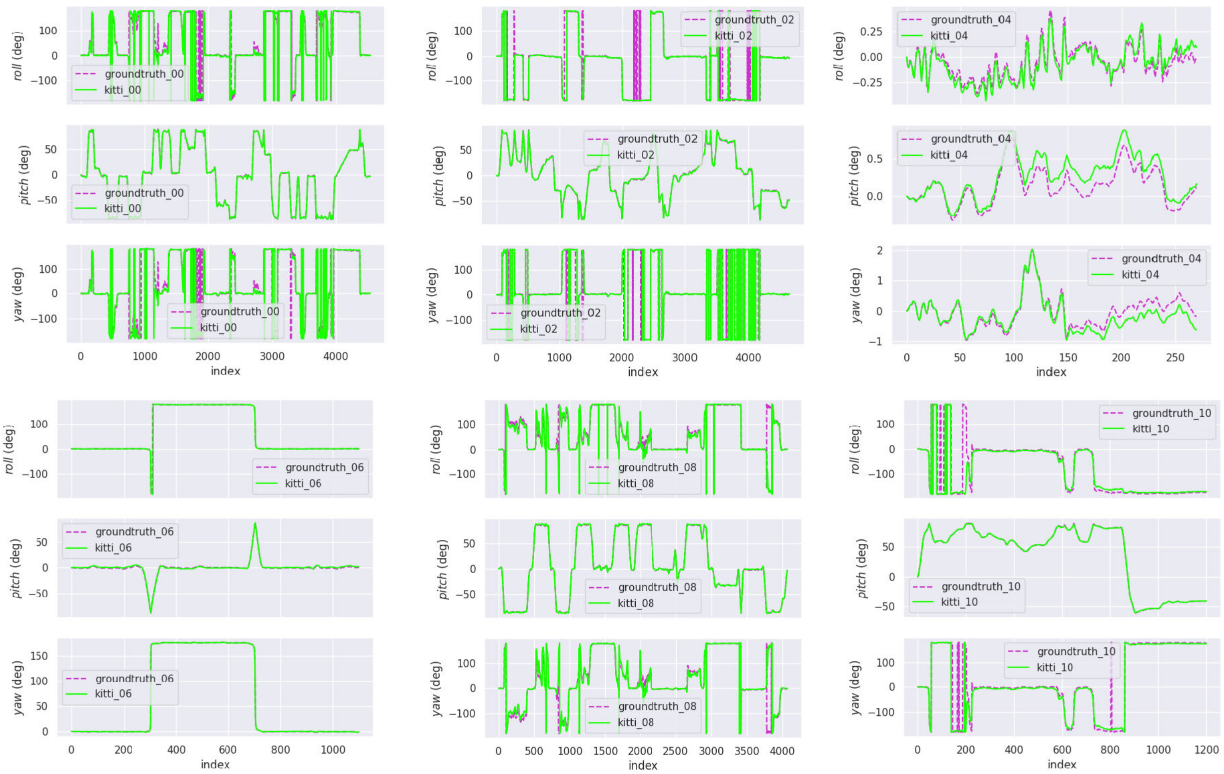


FIGURE 6. The results of pose estimation using the proposed VIO framework on the sequences of (a) kitti_00, (b) kitti_02, (c) kitti_04, (d) kitti_06, (e) kitti_08 and (f) kitti_10, which were mainly used to provide visual reference for further path planning by outputting the vehicle's poses.

TABLE 2. The Variation of searching time, nodes and distance with different numbers of obstacles.

No. of obstacles	Time	Nodes	Distance
500	2.827823	736	1217.229
400	2.588683	730	1217.229
300	2.429026	724	1217.229
200	2.242238	718	1217.229
100	2.147291	716	1217.229
50	2.031933	712	1217.229

A. DATASETS

1) KITTI DATASET

The KITTI dataset is used to assess the stereo and optical flow, visual odometry (VO), 3D object detection and 3D tracking performance of computer vision technologies in vehicle mounted environments. KITTI includes real image data from urban, rural and highway scenes. In the paper, we use only 11 sequences for visual evaluation, which have corresponding ground truth poses obtained by a GPS reader.

2) CITYSCAPES

Cityscapes is a new large-scale automatic driving dataset, which contains a set of different stereo video sequences recording street scenes from 50 different cities with high-quality annotation of 5K frames at the pixel level, except for a set of 2K frames with weak annotation. In our work, we use ''leftImg8bit_trainvaltest'' for performance evaluation.

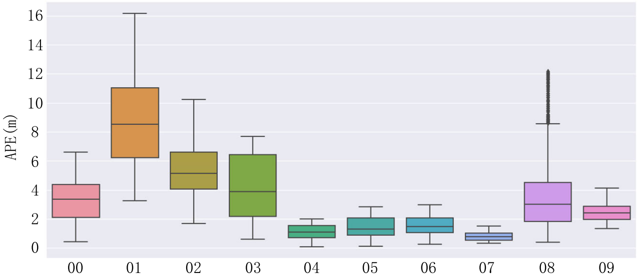


FIGURE 7. The APE of vehicles based on kitti_00 ∼ kitti_10. The box chart represents the change levels of APE for some sequence, which consists of a minimum, lower quartile, median, upper quartile, maximum, mean and interquartile range.

B. POSE ESTIMATION WITH A VIO

Pose estimation needs to monitor the changes of driving direction, which consists of translation and rotation motions. Based on the trajectory of rotation, we can attempt to predict the vehicle's future trajectory to build some references for the control center of intelligent vehicles for path planning. In this paper, we train a deep learning-based VIO model to estimate the pose, as shown in Fig. 6. We use the Euler angle (*yaw*, *pitch* and *roll*) to describe the vehicle's rotation in the motion space. In addition, we also use three separate rotation angles.

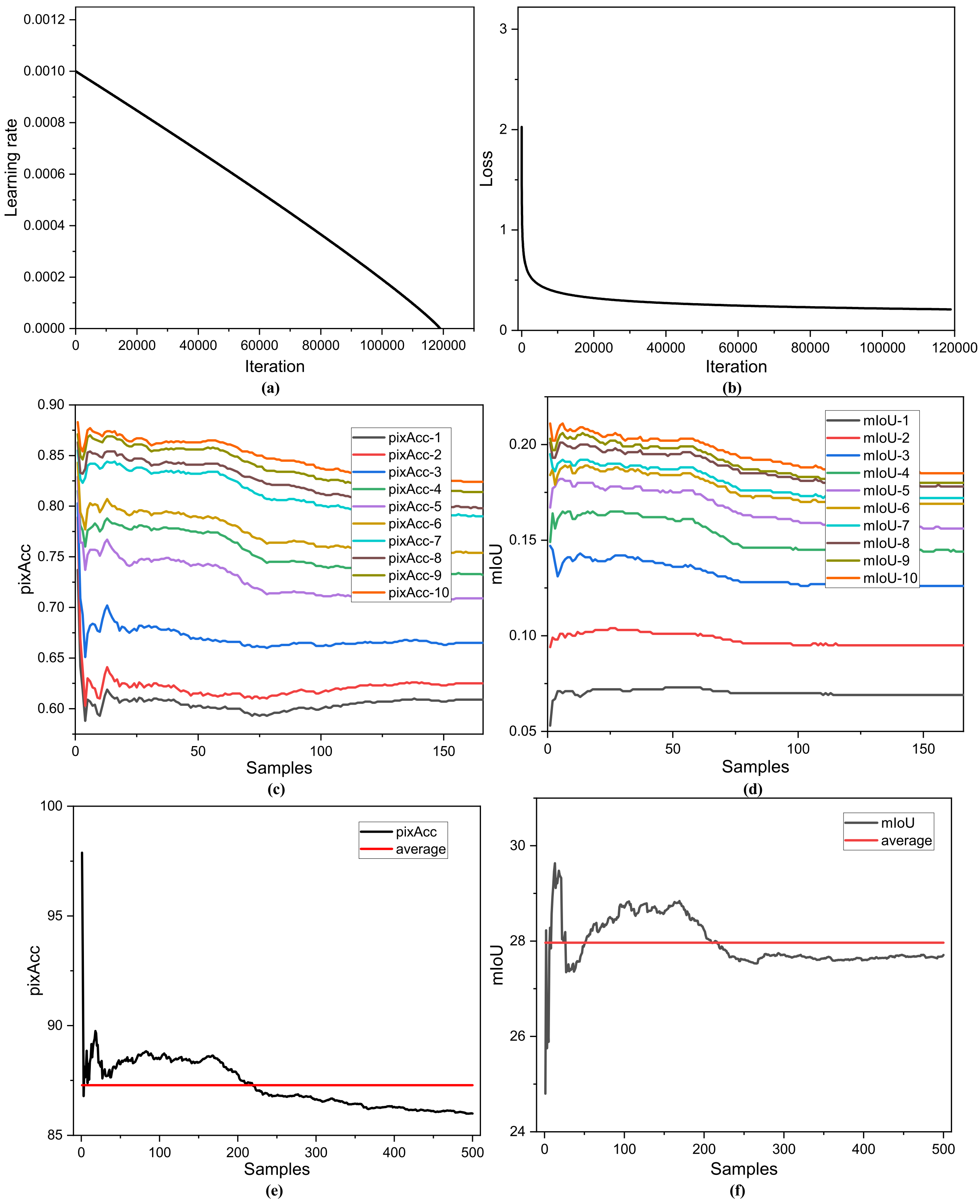


FIGURE 8. The curve of training parameters. (a) and (b) represent the learning rate and loss changes as the number of iterations increases. (c) and (d) demonstrate the accuracy and mIoU changes with the iteration increase. (e) and (f) show the whole training accuracy and mIoU changes with the increase of iteration.

Furthermore, we select the APE (absolute pose error) as the evaluation metric of pose estimation. The APE is often used for evaluating a vehicle's absolute trajectory error. The corresponding vehicle's pose is directly estimated and referenced by the pose relationship. Then, the statistics of the whole trajectory are calculated, which are useful for testing the global consistency of the trajectory. It can be seen from Fig. 7 that the results based on the proposed method on kitti_07 are better than the results of other sequences, and the APE is in the range of 0 ~2. The result on kitti_01 is very poor, and its APE is in the range of 3 ~ 17. On the whole, the APE of the 11 KITTI sequences is distributed mainly in the range of 0 ~ 7, which indicates that the proposed method has reached an advanced level. The results on kitti_01 is poor, mainly due to the many sharp turns on this path that results in instant trajectory drift.

C. SEMANTIC SEGMENTATION FOR OBSTACLE DETECTION

In our work, by learning from E-net [48], we reconstruct a segmentation network. Compared with E-net, the reconstructed network has a deeper layer and a symmetrical down-sample and up-sample structure, which is different from E-net and ensures a better scene segmentation effect to a large extent. In addition, compared with other popular semantic segmentation networks, the proposed network has lightweight parameters (only has 2.3M weights). Therefore, the proposed segmentation network can carry out image segmentation in real time on a small vehicle processor. The experiment is running on an NVIDIA Jetson Xavier NX, which has the same size as the Jetson Nano, but with better performance. In the process of segmentation, the dataset is set as 166 samples for training and 500 samples for evaluation, the epoch is set as 3, and the iteration is up to 118920. It can be seen from Fig. 8 (a) and (b) that as the number of iterations increases and the gradient of network learning rate decreases, the training loss decreases gradually until it approaches 0.

To evaluate the performance of the proposed method, we utilize PixAcc (pixel accuracy) and mIoU (mean intersection over union) as evaluation metrics of segmentation, as shown in Fig. 8 (e) and (f). The results show that the average values of PixAcc and mIoU are 87.3 and 28.1, respectively, and show different PixAcc and mIoU values for each sample. In addition, we also show the changes of accuracy and mIoU of 166 training samples in ten verifications during the first 9900 iterations in Fig. 8 (c) and (d). We can see that the values of accuracy and mIoU for 166 samples increase gradually with the increase of iteration.

We also evaluate the performance of semantic segmentation by comparing the pro-posed method with other state-of-the-art methods, e.g., Fcn8s, Fcn16s, Fcn32s, E-net and Led-net. In our work, the front-end of the network model in the experiment is the above network model in turn, and the back-end takes vgg-16 as the skeleton. We adopt a sigmoid parameterized activation function to encode the cost performance of segmentation, which is able to comprehensively evaluate the accuracy and computational cost. P_{acc} and P_{mIoU} are adopted as the evaluation metrics of cost performance considering precision and mIoU:

$$P_{acc} = \frac{Acc}{sigmoid\left\{(w \times t), \alpha\right\}} \tag{12}$$

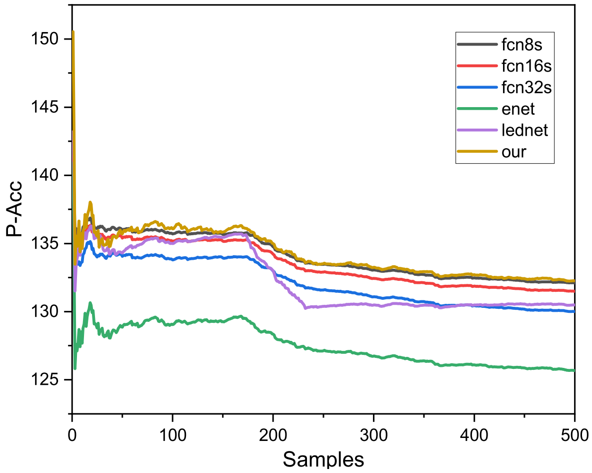


FIGURE 9. The comparison of P_{acc} between the proposed and state-of-the-art.

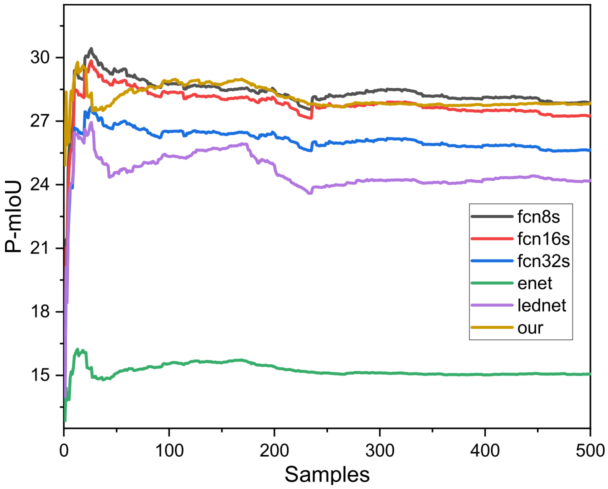


FIGURE 10. The comparison of P_{mIoU} between the proposed and state-of-the-art methods.

$$P_{mIoU} = \frac{mIoU}{sigmoid\left\{(w \times t), \beta\right\}} \tag{13}$$

where Acc and $mIoU$ represent the accuracy and mean intersection of union (mIoU), $sigmoid$ represents the parameterized activation function, and w and t represent the weight and the cost time of an iteration. α and β represent dynamic parameters in the activation function.

Fig. 9 and Fig. 10 show the results of accuracy and mIoU for the proposed method and other popular methods. It can be seen that the performance of the proposed method is better than that of Fcn8s, Fcn16s, Fcn32s and Led-net considering the accuracy or mIoU; its weight and time consumption are slightly lower than that of Fcn8s. The main reason is that after training, the weight of our proposed network model is smaller, approximately one thirtieth that of Fcn8s, Fcn16s, Fcn32s and Led-net, and the iteration speed is faster, although it is slightly lacking in performance.

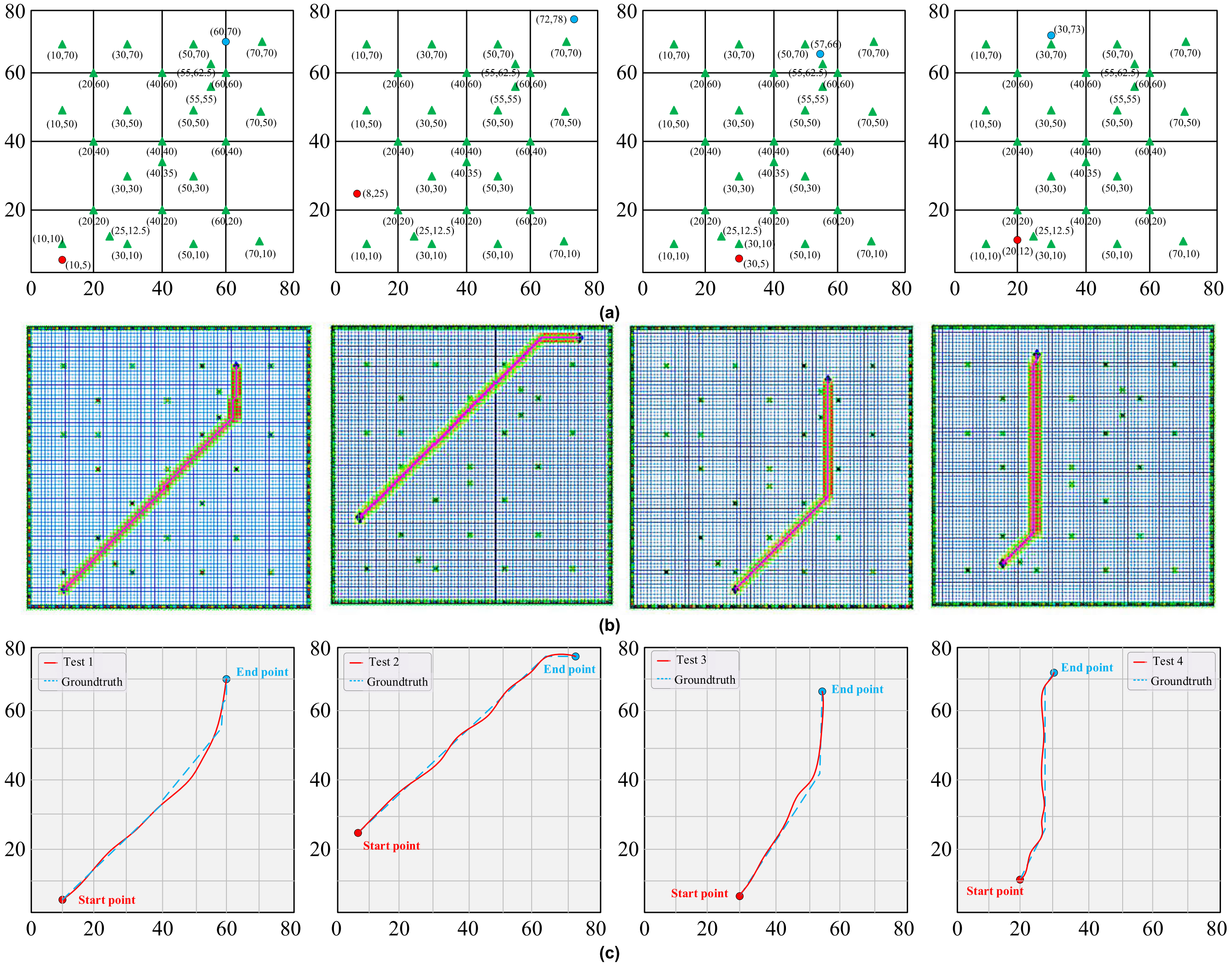


FIGURE 11. Obstacle avoidance and path optimization of an intelligent vehicle with different start-end points. (a) Obstacle locations. We set 27 obstacles and four different start-end points in the same scene. (b) A* path planning. The four different paths show that A* is able to help the vehicle avoid obstacles and determine the best path for the vehicle to drive. (c) Real time path searching. A Jetson device is used as the control center of the vehicle to search for the path in real time.

D. PATH PLANNING FOR OBSTACLE AVOIDANCE

The path planning of vehicles needs to deal with the challenging problems of obtaining safe and effective driving routes for autonomous vehicles. In fact, path planning technology is currently a very hot research topic. What makes path planning so complex is that it covers all areas of autonomous driving technology, from the most basic brakes, to sensors that sense the environment, to locating and forecasting models. To verify the proposed method, we conduct some experiments in simulative scenes. The experimental scene is flat with a length and a width of 8 meters. We place some obstacles in this scene to simulate a real automatic driving experience, as shown in Fig. 11 (a), and take a small car equipped with a Jetson Xavier NX as an automatic driving vehicle. The distribution of obstacles and start-end points is shown in Fig. 11 (b) and Table 3, which present the basic settings, as well as the process of path searching, including searching distance, searching times and searching nodes. Additionally, we run our model on the CPU and GPU, respectively, and compare the time cost in each processing stage, as shown in Table 4. It can be seen that in the stage of image preprocessing and visualization, the CPU is faster than the GPU, but in the stage of network training, the GPU is significantly faster than the CPU. In the stage of image post-processing, the processing speed is similar to that of the GPU. Finally, in terms of total time, the GPU is faster than the CPU.

Fig. 11 (c) presents the comparison between the path trajectory and ground truth, which can be seen that with different starting point and end point setting conditions, the real measurement data and the ground truth are basically consistent, which ensures the safety and success of avoiding obstacles and guarantees that the vehicles reach the target point quickly and efficiently.

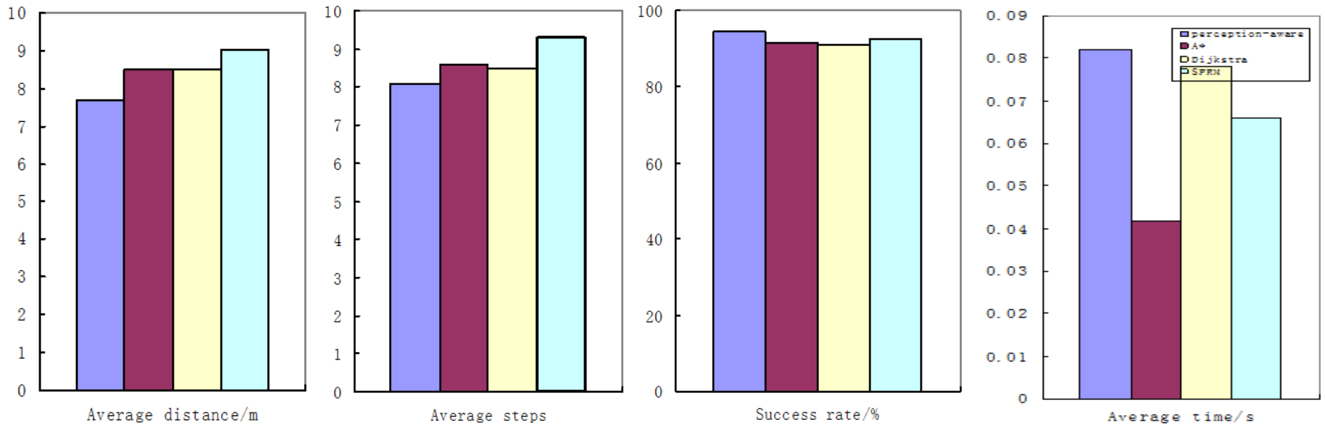


FIGURE 12. Results of path planning on a 7 × 7 × 7 (m) 3D map. The metrics of evaluation consist of average planning distance, average planning steps, success rate and average planning time.

TABLE 3. Path planning results with different start and end points.

Test	Start point/ *dm*	End point/ *dm*	Distance/ *dm*	Computation cost/ *ms*	Nodes
1	(10,5)	(60,70)	85.71068	1.528575	560
2	(8,25)	(72,78)	65.14214	4.199665	872
3	(30,5)	(57,66)	85.95332	4.090234	512
4	(20,12)	(30,73)	72.18377	1.416975	488

TABLE 4. Time consumption of each stage in the process of scene segmentation.

Process	CPU/ *ms*	GPU (CUDA)/ *ms*
Pre-process	0.03956	1.24528
Network	48.42616	46.27104
Post-process	0.76358	0.76250
Visualize	0.03191	0.29104
Total	49.26121	48.56956

E. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS

We randomly generate a 7×7×7 (m) 3D map full of obstacles to evaluate the effectiveness of various existing path planning methods, which include the proposed perception-aware path planning method, A* [51], Disjkstra [52] and SPRM [53], as shown in Fig. 12. On the same map, the proposed perception-aware path planning method gives the smallest search distance and least steps while achieving a higher success rate. However, due to the adoption of the planning method of "scene perception + A*", the total path planning time is composed of image processing time and path searching time. Therefore, the time of path planning is slightly longer than that of other algorithms. Fig. 13 presents results of examples based on the proposed perception-aware path planning method in an outdoor environment.

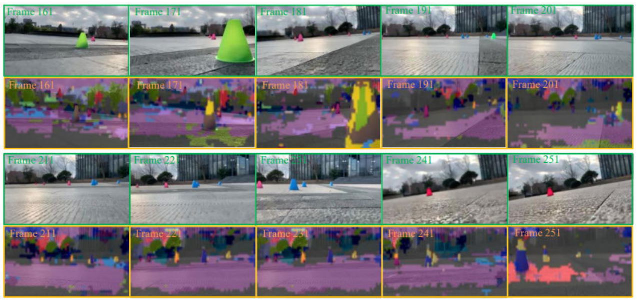


FIGURE 13. Examples of perception-aware path planning. The blue and orange rectangles present the original images and the segmented images, respectively.

V. CONCLUSION

In this paper, the multi-model perception-aware method is proposed to solve path planning problems under complex environment. The proposed method carries out path planning task based on the scene perception (including pose estimation and scene segmentation) and path searching. We first evaluate the proposed pose estimation model by comparing it with the ground truth on the KITTI dataset based on the metric of APE. The mathematical analysis showed that the proposed model is consistent with the ground truth in the view of pose trajectory.

Then, we evaluate the proposed lightweight segmentation network by comparing with state-of-the-art methods based on mIoU. According to the comparison results based on accuracy and mIoU, we conclude the proposed lightweight model obtain a higher obstacle detection accuracy.

Finally, we carry out an experiment in a simulated environment full of obstacles and compare the pipeline with existing path planning methods that do not use scene perception. The results show that the path planning method based on perception-awareness is better than methods that do not refer to scene information in terms of path search time, search distance, search steps and success rate of obstacle avoidance.

Although the above advantages are presented, there are still some problems that have not been solved. For example, we have only verified the method in a simulation environment (static environments) and not in a real, complex environment.

The proposed path planning method is very dependent on the model design, so vehicles do not have the ability to learn actions independently. Therefore, under extreme driving conditions, the proposed method may be extremely unstable, especially in a dynamic environment. Therefore, in the future, we will focus on the algorithm design of pose learning for intelligent vehicles and optimal design of engineering problems [54]–[56].

In addition, to deal with the path planning of dynamic environments, a deep reinforcement learning (DRL) model will be presented in the future work. In the process of outputting the pose and path, the vehicles can learn this behavior online and constantly optimize the path, which will make the vehicles more intelligent and autonomous as time goes on to meet the needs of future driverless vehicles under dynamic scenes.

REFERENCES

[1] J. Zheng, S. Yang, X. Wang, X. Xia, Y. Xiao, and T. Li, "A decision tree based road recognition approach using roadside fixed 3D LiDAR sensors," *IEEE Access*, vol. 7, pp. 53878–53890, 2019.

[2] J. Wu, H. Xu, Y. Sun, J. Zheng, and R. Yue, "Automatic background filtering method for roadside LiDAR data," *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 106–114, 2018.

[3] J. Guo, M.-J. Tsai, and J.-Y. Han, "Automatic reconstruction of road surface features by using terrestrial mobile LiDAR," *Autom. Construct.*, vol. 58, pp. 165–175, Oct. 2015.

[4] K. Geng, G. Dong, G. Yin, and J. Hu, "Deep dual-modal traffic objects instance segmentation method using camera and LiDAR data for autonomous driving," *Remote Sens.*, vol. 12, no. 20, p. 3274, Oct. 2020.

[5] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler, "Project AutoVision: Localization and 3D scene perception for an autonomous vehicle with a multi-camera system," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4695–4702.

[6] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image Vis. Comput.*, vol. 68, pp. 14–27, Dec. 2017.

[7] H. Masuta, Y. Okajima, K. Sawai, T. Motoyoshi, T. Tamamoto, K. Koyanagi, and T. Oshima, "Visual perception of approaching object using spherical camera," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 2254–2259.

[8] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Perez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9308–9318.

[9] B. Barrois, S. Hristova, C. Wohler, F. Kummert, and C. Hermes, "3D pose estimation of vehicles using a stereo camera," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2009, pp. 267–272.

[10] J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano, and D. Wooden, "Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 619–626.

[11] M. K. Kaiser, N. R. Gans, and W. E. Dixon, "Vision-based estimation for guidance, navigation, and control of an aerial vehicle," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 3, pp. 1064–1077, Jul. 2010.

[12] A. D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A. López, "An efficient approach to onboard stereo vision system pose estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 476–490, Sep. 2008.

[13] D. Aufderheide, W. Krybus, and G. Edwards, "VIFTrack!–visual-inertial feature tracking based on affine photometric warping," in *Computational Vision and Medical Image Processing IV: VIPIMAGE*. Funchal, Portugal: CSC Press, 2014.

[14] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "MAPLAB: An open framework for research in visual-inertial mapping and localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, Jul. 2018.

[15] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1974–1982.

[16] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation," 2017, *arXiv:1708.00894*. [Online]. Available: http://arxiv.org/abs/1708.00894

[17] A. Rajagopal, G. P. Joshi, A. Ramachandran, R. T. Subhalakshmi, M. Khari, S. Jha, K. Shankar, and J. You, "A deep learning model based on multi-objective particle swarm optimization for scene classification in unmanned aerial vehicles," *IEEE Access*, vol. 8, pp. 135383–135393, 2020.

[18] C. Tao, L. Mi, Y. Li, J. Qi, Y. Xiao, and J. Zhang, "Scene context-driven vehicle detection in high-resolution aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7339–7351, Oct. 2019.

[19] X. Yan, S. Wang, X. Liu, Y. Han, Y. Duan, and Q. Li, "Infrared image segment and fault location for power equipment," *J. Phys., Conf. Ser.*, vol. 1302, no. 3, Aug. 2019, Art. no. 032022.

[20] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5391–5399.

[21] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.

[22] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[23] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–8.

[24] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, 2017.

[25] F. Huang, A. Zeng, M. Liu, Q. Lai, and Q. Xu, "DeepFuse: An IMU-aware network for real-time 3D human pose estimation from multi-view image," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 429–438.

[26] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints," 2019, *arXiv:1906.11435*. [Online]. Available: http://arxiv.org/abs/1906.11435

[27] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 1–7.

[28] Y. Almalioglu, M. Turan, A. E. Sari, M. Risqi U. Saputra, P. P. B. de Gusmão , A. Markham, and N. Trigoni, "SelfVIO: Self-supervised deep monocular visual-inertial odometry and depth estimation," 2019, *arXiv:1911.09968*. [Online]. Available: http://arxiv.org/abs/1911.09968

[29] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[30] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[31] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 648–657.

[32] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 289–305.

[33] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4151–4160.

[34] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12607–12616.

[35] Y. Ma, M. Hu, and X. Yan, "Multi-objective path planning for unmanned surface vehicle with currents effects," *ISA Trans.*, vol. 75, pp. 137–156, Apr. 2018.

[36] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, "Applying Dijkstra algorithm for solving neutrosophic shortest path problem," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Nov. 2016, pp. 412–416.

[37] G. Qing, Z. Zheng, and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 7138–7143.

[38] M. Dehghani, Z. Montazeri, G. Dhiman, O. P. Malik, R. Morales-Menendez, R. A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero , and L. Parra-Arroyo, "A spring search algorithm applied to engineering optimization problems," *Appl. Sci.*, vol. 10, no. 18, p. 6173, Sep. 2020.

[39] X. Wang, Y. Shi, D. Ding, and X. Gu, "Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning," *Eng. Optim.*, vol. 48, no. 2, pp. 299–316, Feb. 2016.

[40] H. N. Ghafil and K. Jármai, "Dynamic differential annealed optimization: New Metaheuristic optimization algorithm for engineering applications," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106392.

[41] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020.

[42] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi, "Path planning for active SLAM based on deep reinforcement learning under unknown environments," *Intell. Service Robot.*, vol. 13, pp. 263–272, Jan. 2020.

[43] X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, "Learn to navigate: Cooperative path planning for unmanned surface vehicles using deep reinforcement learning," *IEEE Access*, vol. 7, pp. 165262–165278, 2019.

[44] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156787–156803, 2019.

[45] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput. J.*, vol. 77, pp. 236–251, Apr. 2019.

[46] R. Kala, A. Shukla, and R. Tiwari, "Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 307–327, Apr. 2010.

[47] X. Zhong, J. Tian, H. Hu, and X. Peng, "Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *J. Intell. Robotic Syst.*, vol. 99, no. 1, pp. 65–77, Jul. 2020.

[48] M. Faria, R. Marín, M. Popović, I. Maza, and A. Viguria, "Efficient lazy Theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV," *Sensors*, vol. 19, no. 1, p. 174, Jan. 2019.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[50] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: http://arxiv.org/abs/1606.02147

[51] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Proc. Eng.*, vol. 96, pp. 59–69, Jan. 2014.

[52] M. Enayattabar, A. Ebrahimnejad, and H. Motameni, "Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment," *Complex Intell. Syst.*, vol. 5, no. 2, pp. 93–100, Jun. 2019.

[53] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artif. Intell.*, vol. 219, pp. 1–24, Feb. 2015.

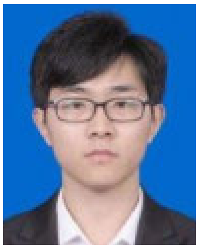
[54] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl. Based Syst.*, vol. 159, pp. 20–50, Nov. 2018.

[55] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.

[56] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts game optimizer: A new optimization technique based on darts game," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 286–294, Oct. 2020.

ZHENYU LI (Student Member, IEEE) received the M.S. degree in mechanical engineering from the Shandong University of Technology, Zibo, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Mechanical Engineering, Tongji University, Shanghai, China. His research interests include cover deep learning, scene perception and autonomous localization, and navigation for autonomous driving.

AIGUO ZHOU received the Ph.D. degree from the Shanghai Jiaotong University of Mechatronics Engineering, in 2004. He is currently an Associate Professor with Tonji University, Shanghai China. In 2010, he was a Visiting Scholar with Okland University, Michigan, USA. He has over 30 publications, including conference and journal papers. His research interests include smart sensor, intelligent vehicle with a focus on system design, and control strategy.



JIAKUN PU received the B.S. degree in mechanical engineering from Tongji University, in 2019, where he is currently pursuing the master's degree. His research interests include embedded development and information security.

JIANGYANG YU received the B.S. degree in mechanical engineering from Tongji University, in 2019, where he is currently pursuing the master's degree. His research interest includes the definition and implementation of visual detection algorithms.