Autonomous Navigation for Mobile Robots with Weakly-Supervised Segmentation Network

Peinan Huang
Dept of Automation
Shanghai JiaoTong Uuniversity
Shanghai, China
lyrics@sjtu.edu.cn

Jialun Li
Dept of Automation
Shanghai JiaoTong Uuniversity
Shanghai, China
jialunli@sjtu.edu.cn

Jianping He
Dept of Automation
Shanghai JiaoTong Uuniversity
Shanghai, China
jphe@sjtu.edu.cn

***Abstract*—Low-cost sensors are promising to increase the use of service robots. This paper investigates autonomous navigation for mobile robots with a low-cost monocular camera. The main challenge lies in two parts. First, how to segment drivable areas in prior unseen environment without depth information of the camera. Second, how to generate safe and flexible path for the robot to track with changing and even unstable visual information. A semantic segmentation network is proposed, which takes the image of monocular camera as input and outputs the drivable area. The network is trained with 2D-lidar labelled data, known as the weakly-supervised method. To reduce segmentation noise and improve accuracy, we establish a probability occupancy map according to the distance between robot and obstacles. For path generation, we project obstacles from image space into local Frenet frame, and present a novel local planning approach to smooth and stitch paths with consecutive unstable image inputs. Simulation results show feasibility and robustness of our method.**

***Index Terms*—Low-cost Robots, Semantic Segmentation, Path Planning**

I. Introduction

In recent years, mobile robots have been developing rapidly all over the world. Currently most indoor robots mainly depends on lidar sensors for navigation. At the same time, the cost of these sensors are still high enough and cannot be used on a large scale, which is one of the major reasons that prevents the widespread use of autonomous robots.

With rapid development of deep learning, monocular vision based methods for perception and planning are considered a feasible and complex solution to achieve human level performance at low cost [1] [2]. Vision-based obstacle avoidance for robot has attracted a lot of research. Semantic segmentation is a vision-based method and usually undertakes the task of scene understanding. For indoor robot, we propose a framework combining semantic segmentation and path planning to realize pure visual-based obstacle avoidance, which provides a new perspective for autonomous indoor robot navigation. The most unique aspect of our approach is that the results of semantic segmentation are transformed into constraints for robot path planning.

For indoor mobile robots, using 2D-lidar and camera data, we adopt a weakly supervised method to train semantic segmentation network based on [3], but extend it to indoor environments, fix the problem of discontinuous labeling of obstacles and divide the environment into drivable and non-drivable regions. In the deployment stage, the trained segmentation network is used to obtain the key obstacle information and map the obstacle points to the world frame. To reduce noise from inaccurate segmentation, we construct a probability occupancy map by using obstacle positions and their distance. In the planning step, we design a search problem to find constraint variables from obstacles. The planning module then carry out local path planning based on an improved quadratic optimization method to avoid obstacles and reach the target location.

To evaluate our navigation scheme, We test our segmentation and planning framework in different indoor simulation scenes. We introduce one-pixel error to do segmentation error analysis. The results show that the segmentation network can meet the accuracy requirements of obstacle avoidance in indoor scenario.

II. Related work

For mobile robots, the premise of obstacle avoidance is to perceive the position of obstacles in the surrounding environment. In order to use camera to obtain obstacle information, obstacles need to be detected at pixel level, which is the task of segmentation. Considering the variety of obstacles, a common method is road segmentation, which is to get the area that car or robot can pass through. The goal of road segmentation is to label the drivable region, also called free space. At present, many works have been done to segment drivable regions from images. Jian Yao et al. estimated the driveable collision free space from monocular video by reducing the free space estimation task to an inference problem on a 1D graph, designed a potential function and used SVM to learn parameters [4]. Morten Rufus Blas used compact color and texture descriptor and K-means to achieve segmentation for outdoor Robots [5].

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China. E-mail: {lyrics, jialunli, jphe}@sjtu.edu.cn.

Deep semantic segmentation network has developed continuously in recent years, showing a good segmentation performance and gradually become the mainstream [6]. Semantic segmentation is usually used for scene understanding, still there have been some work on road segmentation based on deep semantic segmentation network. Zhidong Deng proposed an RPP model for monocular vision-based road segmentation based on the combination of fully convolutional network, residual learning, and pyramid pooling [7]. In order to train a deep segmentation network, we usually need a large-scale dataset with hand-labeled ground truth. However, generating such a dataset is time-consuming and labor-intensive. To solve the problem of manual labeling, Wang et al. used classic semantic segmentation Neural Network like FuseNet and based on RGB-D data to automatically segment the driveable area and road foreign objects [8]. Wang X performed weakly supervised semantic segmentation in autonomous driving scenes based on image tags and clustering [9]. Barnes D et al. proposed a weakly supervised method for segmentation from images [3]. Their method, which divides the image into three parts: driving path, obstacle and driveable area, is tested in outdoor road scene.

However, the relatively low accuracy and noise of semantic segmentation is still a problem, especially for segmenting obstacles. R. Miyamoto, M Showed Using appropriate datasets with mutil classes can improve accuracy of semantic segmentation for robot navigation [10], which showed the feasibility of using segmentation information for robot navigation. Ryusuke Miyamoto et al. used the monocular camera for visual navigation and determined the target point according to the result of semantic segmentation [1]. but their method is used in outdoor environment, the complexity of semantic segmentation leads to its lack of robustness. At the same time, the method of finding target point directly from the image can not achieve fine path planning.

In thie paper, we apply semantic segmentation to indoor robot navigation, because under outdoor scenario, the ground texture features change dramatically, which is difficult to segment accurately. The ground is more structured and smooth in indoor scenes. And the positioning accuracy of segmentation will decrease with the increase of distance. The distance of objects in indoor scene is relatively close, which is conducive to improve the navigation accuracy. At the same time, the ground in outdoor environment usually has slope. For indoor environment, however, the horizontal ground can be used as the constraint of obstacle location mapping. Here we design a framework integrating methods in [3] with a path planning module to achieve obstacle avoidance indoors. Meanwhile we use obstacle positions and their distance to construct a probability occupancy map to reduce noise from inaccurate segmentation.

Obtaining the surrounding information from image is for path planning. The requirements of path planning include avoiding obstacles, smoothness, shortest path, meeting the dynamic characteristics of the robot and so on. It can be roughly divided into the following four categories: search based methods, sampling based methods, interpolation based methods and optimization based methods [11]. The method based on optimization can output smooth paths, and its robustness is relatively good. Yajia Zhang formulate the path optimization problem as a quadratic programming [12]. But they assumed constraints information have already been given by sensors. In this work, we will base our planning algorithm on [12] and show how to get the obstacles constraints from image and apply it for indoor robots navigation.

III. Problem Description

The overall problem considered in this paper is to realize reliable path planning based on monocular vision in common indoor environments (such as office or household buildings). Specifically, the spatial coordinates $(x^*,\ y^*)$ of target point is given. The robot moves from the initial position $(x_0,\ y_0)$ to the target point, while realizing obstacle avoidance. Our assumptions include that only static obstacles are considered, and there are priori information of spatial indoor layout for global reference path planning, which can be easily obtained through architectural drawings or building a map by SLAM in advance, and is stored in the form of grid map. But we don not need high precision and resolution. The robot is required to plan a path, avoid possible unknown obstacles and reach the target location smoothly.

IV. Method

A. System Overview

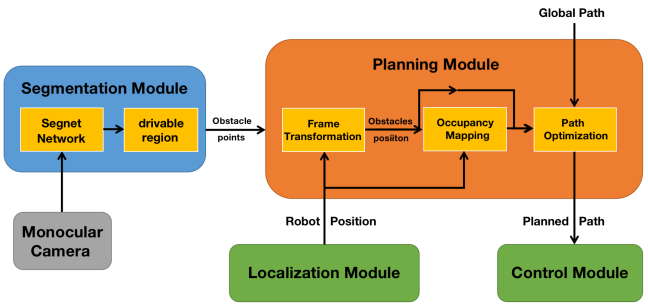


Fig. 1. Schematic representation of the proposed robot navigation framework.

As shown in the Fig. 1, the proposed robot navigation framework consists of four main modules: segmentation, planning, localization and control. The segmentation module get the input image of a monocular camera, and segment drivable and non-drivable regions through a deep segmentation network, so as to obtain the location of key obstacle points, which are at the junction of objects and ground. The path planning module obtains the spatial position of the obstacle points through coordinate transformation, and establishes an occupancy grid map, given the robot pose information obtained by localization module. Then, the path is optimized based on

the pre-planned global reference path, occupancy grid map and obstacle coordinates. Finally, the planned path is sent to the control module for trajectory tracking and obstacle avoidance. In the following, we mainly focus on the segmentation and planning modules.

B. Network Training

1) Data Acquisition and Annotation: The mobile robot is equipped with a 2D lidar sensor and a monocular camera. In the indoor environment, the robot moves with a certain strategy (according to the preset trajectory in the simulation environment, manually controlled or randomly move in the real environment), and collects the time-aligned image data and lidar data.

The position of the obstacle is mapped to the image by using the laser data, and the image is labeled pixel by pixel based on the method in [3]. The original method has three segmentation categories: driveable, obstacle and path proposal. We use drivable and non-drivable classification, that is, each pixel belongs to either region where the robot can go through or obstacle region. The specific labelling process is as follows.

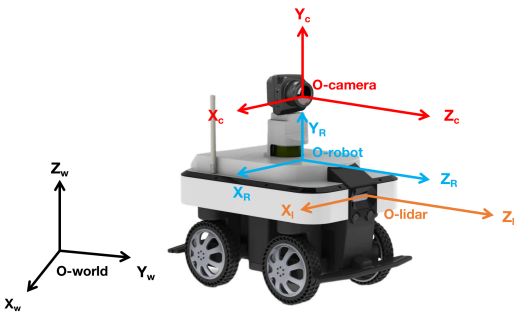


Fig. 2. Four frames established on the robot.

There are five coordinate frames including image frame F_{image}, world frame F_w, camera frame F_c, robot frame F_r and laser frame F_l. The o-xy planes of world, robot and laser frames are parallel to the ground, and there are no special requirements for camera frame. In order to help representation, the origin of the world frame is set on the ground. The height of the laser sensor is set to h_l. The obstacle point information obtained by 2D lidar is represented by a list, and the obstacle point position data (d, θ), where d is the distance to the sensor, and θ is the corresponding angle. It is easy to convert them into the coordinates P_l = (X_l,Y_l,Z_l) in the laser frame, where Y_l = - h_l.

Before the lidar senor collects data, horizontal calibration is required, and the ground of the indoor environment is assumed to be horizontal. Therefore, the height of the scanned obstacle point is fixed value h_l. Convert the coordinates of the obstacle point to the camera frame as P = (X_c,Y_c,Z_c). Based on the pinhole camera model, the obstacle point coordinates (u, v) in the image frame can be obtained by using the following equation.

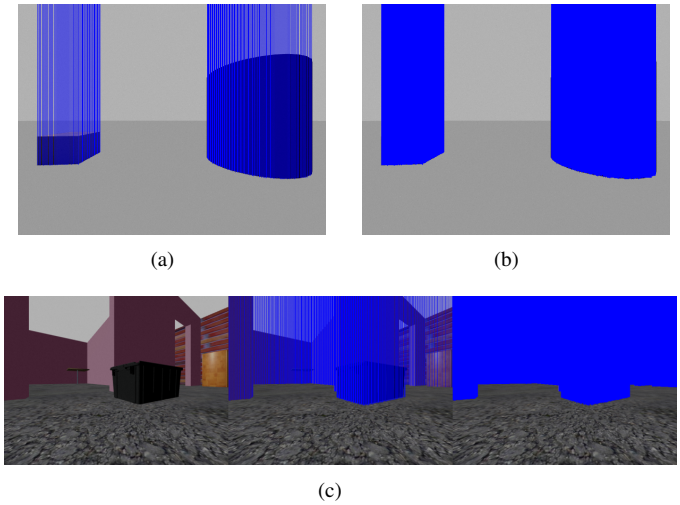
$$P = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R(\begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} - C)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \triangleq \frac{1}{Z_c} KP \qquad (1)$$

where (R, C) is the external parameters from laser to camera frame calibrated by the monocular camera and K(C_x, C_y, f_x, f_y) is the internal parameter matrix.

Distortion has a great impact on the mapping. Generally, wide-angle lens has large distortion, and usually we don't need too large angle of view (AOV) to reduce distortion. For the calibrated camera, we also need to correct the distortion to eliminate the radial distortion and tangential distortion in the image.

Label the obstacle pixels mapped to the image and all pixels above the same column of the point as non-drivable as in [3]. A typical example result is shown in the Fig. 3. Due to the difference between the resolution of laser sensor and the image, some points of the same object labeled on the image may be discontinuous. Therefore, we design a method to guarantee continuity. A sliding window is used to scan twice from left to right. 1 represents that there is an obstacle in the column and 0 represents no obstacle in the column. When there are columns satisfying discontinuous mode in the window (eg, 1101, 1011), interpolation is used to complete.



(a) (b)

(c)

Fig. 3. (a) is an example of image labeling obstacle points with discontinuous columns of images. (b) is image labeling after continuity check. (c) is comparison between the original image obtained by the camera and the image after automatic annotation.

2) Semantic Segmentation Network Training: Supervised Semantic Segmentation can be mainly divided into two categories: region-based and FCN-based semantic segmentation [6]. Common semantic segmentation networks include fcn32, U-net, SegNet, etc. The classic SegNet network is used in [3]. Considering the structure complexity and segmentation

accuracy of the network, we also choose SegNet network. Its biggest feature is that the sampling position (indice) is recorded during down sampling for upsampling image reconstruction. The trained network can segment the drivable areas pixels pure vision based. And obstacle information will be used for subsequent path planning module.

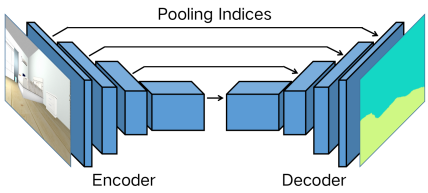
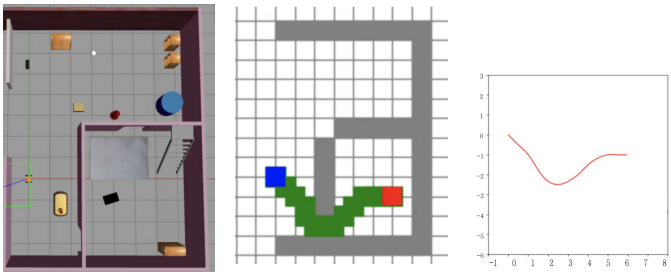


Fig. 4. Segnet network architecture diagram.

C. Path Planning

1) Global Path Generation: Indoor robot global path planning needs to complete the motion between the initial point and the target point. The spatial coordinates of target point in the world frame p_{goal}, the initial position P_0 of the robot and a priori map of the indoor layout with 1m resolution are given. Firstly, A-star algorithm is used to generate coarse global planning path $\{ p_0, p_1 ... p_{goal} \}$. Ensuring smoothness is the premise of planning under Frenet frame later, so the path points are smoothed by mean filtering to obtain $\{ p_0, p_1, ... p_{goal}^{'} \}$. Then a cubic spline curve is generated from the path points as reference. An example is shown in the Fig. 5.



(a) (b) (c)

Fig. 5. (a) indoor environment in simulation (b) path points generated by A-star algorithm (c) cubic spline curve reference path generated

2) Obstacle Information Acquisition: Use monocular camera to collect images in a certain frequency, and use semantic segmentation network to obtain drivable areas. In column by column scanning, the pixels at the junction of drivable and non-drivable areas are recongized as key obstacle points.

The coordinates (u, v) of the obstacle point in image coordinate frame are obtained and mapped to the world frame to obtain the coordinates (X_w, Y_w, Z_w). It is usually not feasible to convert from image coordinates to spatial coordinates, because the image itself is the projection of space on the plane, which compresses the dimensional information. However, since we assume that in indoor scenes the ground is level and the ideal position of obstacle points are the junction of the ground and the obstacle or wall, their height Z_w in the world coordinate frame can be assumed to be zero. Therefore the inverse coordinate transformation has extra plane constraints, that is, the obstacle point is located in the horizontal ground plane, which satisfies the plane equation: $aX_w + bY_w + cZ_w = d$. Let (R, C) be the external parameters from world to camera frame calibrated by the monocular camera, we can get the transformation equations.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \triangleq \frac{1}{Z} KR(\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - C) \tag{2}$$

$$aX_w + bY_w + cZ_w = d$$

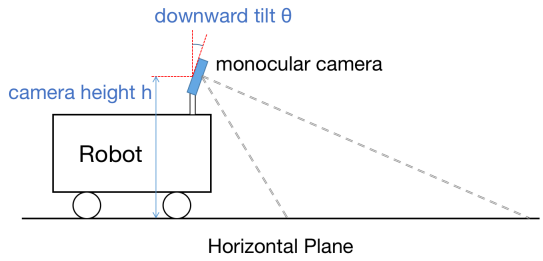


Fig. 6. Pose of monocular camera installed on robot.

Generally, the camera is placed horizontally, so the X axis in the camera frame is parallel to the ground. Note that different camera pose will lead to different field of view and a wider field of view with fewer blind area is preferable. Set the height of the camera to the ground as h, and the downward tilt angle between the Z axis of the camera frame and the ground as θ. Fig. 6 shows a side view of the robot with the camera installed. Then the closed form solution can be obtained as Eq.(3).

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R^{-1} \begin{bmatrix} \frac{u-c_x}{f_x} \frac{f_y}{\cos\theta(v-c_y)+\sin\theta f_y} h \\ \frac{v-c_y}{\cos\theta(v-c_y)+\sin\theta f_y} h \\ \frac{f_y}{\cos\theta(v-c_y)+\sin\theta f_y} h \end{bmatrix} + C \tag{3}$$

When the o-xy plane of the world frame is parallel to the o-xz plane of the camera frame, the constraint can be simplified to $Y_c = constant$ for simple solution.

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R^{-1} \begin{bmatrix} \frac{u-c_x}{v-c_y} \frac{f_y}{f_x} h \\ h \\ \frac{f_y}{v-c_y} h \end{bmatrix} + C \tag{4}$$

The localization module obtains the pose of the robot. Unlike outdoor driving environment, GPS positioning is unreliable indoors, but the pose estimation of the robot itself can depend on other sensor data and fusion methods. Many

localization systems have already been explored for effective and accurate self localization of indoor mobile robots. For example, fusion of odometer and IMU [13], UWB [14], visual SLAM, wireless sensor network, etc. Here, we assume that reliable pose estimation of the robot has been obtained. Then, using the pose information of the robot, the coordinates of the obstacle points are mapped to P(X_w, Y_w, Z_w) in the world frame. According to the global reference path, the positions of the obstacle point and the robot are mapped to the Frenet coordinate, and the path planning is carried out in the Frenet frame.

3) Establishment of Probabilistic Occupancy Map: There may be inaccuracy or error in the obstacles obtained from semantic segmentation. Thus we maintain a discretized global probability occupancy grid map. For a point, P(s = 1) represent the probability that the point is in free state while P(s = 0) represent the probability that it is in occupied state. The ratio of the two probability is introduced as the state of the point: Odd (s) = $\frac{P(s=1)}{P(s=0)}$. The Eq.(5) can be deduced according to the Bayesian formula [15].

$$\log Odd(s|z) = \log \frac{p(z|s=1)}{p(z|s=0)} + \log Odd(s) \tag{5}$$

We update the probability value considering history data. Measurement model is the increment size for update, we think it is related to distance. Because the farther away from the robot, the lower the reliability of the obstacle point, which can be seen in the error analysis discussed later. Hence we set it to be inversely proportional to the distance. As in Eq.(6) and Eq.(7), where dis is the distance from robot to obstacle.

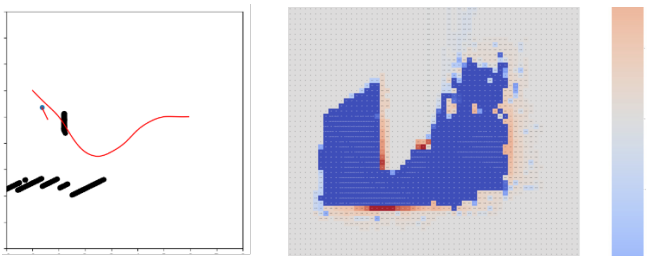
$$\log \frac{p(z=1|s=1)}{p(z=1|s=0)} = \frac{K_1}{dis} \tag{6}$$

$$\log \frac{p(z=0|s=1)}{p(z=0|s=0)} = -\frac{K_2}{dis} \tag{7}$$

By Bresenham algorithm and Eq.(5), the occupation probability of grids on the connecting line between the obstacle and the robot is updated continuously. Only when the probability occupancy exceeds a certain threshold will the robot believe that the obstacles obtained by semantic segmentation are credible. This also reduces the calculation time for next planning. An example of obstacles information acquisition and probability occupancy map establishment is shown in Fig. 6.

4) Generation of Raw Path: The path planning method is based on Baidu Apollo piecewise-jerk optimization algorithm [12], which decouples the trajectory planning into path and speed planning to reduce the computational complexity. For path planning, the numerical optimization is used to solve the optimal path. The optimization problem is carried out in Frenet frame.

To establish and solve the optimization problem, the upper and lower constraint boundaries (l^i_{min}, l^i_{max}) need to be determined firstly. Along s axis of the reference, use ΔS to discretize the s direction equidistant. Set the planned path



(a) (b)

Fig. 7. (a) obstacles (black points) detected in world frame (b) a probability occupancy grid map bulit

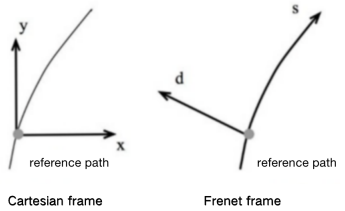


Fig. 8. Cartesian frame and Frenet frame.

length L, then the path planning is equivalent to determine the lateral distance l_i of n path points.

Based on the position information of obstacle points, the upper and lower constraints (l^i_{min}, l^i_{max}) satisfied by l_i can be determined. In order to judge whether an obstacle point is an up or down constraint, from the lateral l dimension perspective, that is, whether to go left or right to avoid obstacle, we propose a search method to get a rough path $\{ (s_{r1}, l_{r1}), ..., (s_{rn}, l_{rn}) \}$. Discrete search, step size L is set to 0.2m. The cost function selects the weighted sum of the distance from the point to the reference line, the shortest distance to the obstacle points for safety consideration, and the distance to the previous search point for smoothness consideration. As in Eq.(8), C_i is the cost between point (s_{ri}, l_{ri}) and (s_{ri-1}, l_{ri-1}), d_{minobs} is the shortest distance from point (s_{ri}, l_{ri}) to all obstacle points and d_0 is a safe distance.

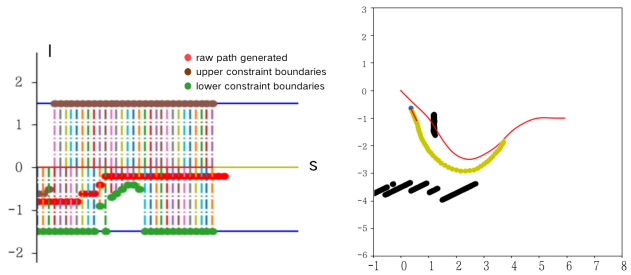
$$C_i = w_1 \times |l_i| + w_2 \times max\{-d_{minobs}, -d_0\} + w_3 \times |l_i - l_{i-1}| \tag{8}$$

5) Path optimization: Starting from each raw path point (s_{ri}, l_{ri}), expand the search upward and downward respectively. When the minimum distance between a point and all obstacle points is less than a certain threshold, this point is considered as the boundary constraint point, then the upper and lower constraints are obtained. Here, we simplify the robot to one point. In order to avoid collision, a buffer distance is reserved when calculating the constraints. At the same time, the constraints l_0, $l_0^{'}$ of the initial state of the planning curve are set according to the current position and orientation of the robot.

$$f = w_l \sum_{i=1}^{N-1} l_i^2 + w_{l'} \sum_{i=1}^{N-1} l_i^{'2} + w_{l''} \sum_{i=1}^{N-1} l_i^{''2} + w_{l'''} \sum_{i=1}^{N-2} l_i^{'''2} \quad (9)$$

$$s.t. \quad l_i \in (l_{min}^i, l_{max}^i)$$

On this basis, path planning is carried out. The optimization variable is the lateral offset l_i of each discrete point, the first and second derivatives $l_i^{'}$ and $l_i^{''}$ on s axis. And the constraints are set to make the third derivative constant. The objective function is as Eq.(9). Set appropriate weights to meet the requirements of being close to the reference line and smoothness. In simulation experiment, we found that if the weight changes dynamically with the initial position of the robot and obstacle constraints, the effect can be better. If the robot deviates greatly from the reference line and the obstacle is far from the reference line, w_l can be increased appropriately to make the planned path closer to the reference line faster.



(a) (b)

Fig. 9. (a) raw path(red points), upper and lower constraint boundaries(brown and green points) generated in simulation. (b) planned path (yellow points) after optimization.

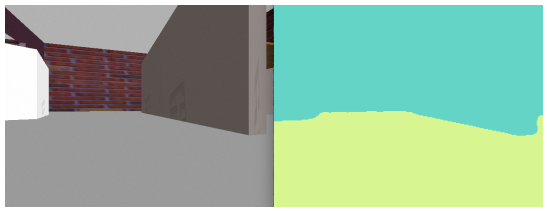
After the optimization problem is established, the osqp library (operator splitting solver for quadratic programs) is used for iterative numerical solution. The solution result is the planned path $\{(s_{opt1}, l_{opt1}), ..., (s_{optn}, l_{optn})\}$. Finally, the path is converted to the world coordinate frame, which will be provided to the control module for trajectory tracking. Linear quadratic Regulator or model predictive control can be used for control policy.

V. Result

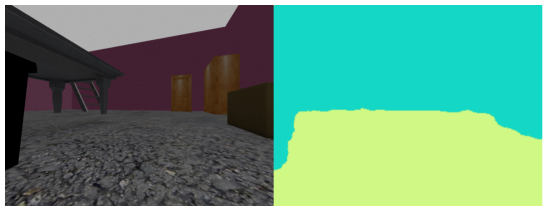
To confirm the feasibility of the proposed scheme, typical indoor environments containing various obstacles are built in Gazebo simulation. Our test environment contains many obstacles that have not appeared in the training process and the angular resolution of lidar is 0.25 degrees. We analyze different semantic segmentation network, including Segnet, FCN, Unet, etc. We use tensorflow to build network and labelled about 300 images each for training. We manually annotated some datasets as true values for testing. Average run time and meanIoU are used as comparison criteria. The calculation speed of 640 x 480 input image is less than 250ms per frame on ordinary CPU computation device. When run on a GPU, the time is less than 100 ms per frame, which is almost real time and practical for implementation.

	Segnet	unet	fcn32
Average Run Time (ms)	200	250	¿600
meanIoU (%)	99.3	99.5	96.0

We then tested the robustness of trained semantic segmentation network. Choose different environments, including simple interior room, textured blanket environment and typical household building with different lighting conditions. As shown in Fig. 9. the segmentation and planning results show that our method performs well in various environments.



(a) environment 1 segmentation result



(b) environment 2 segmentation result

(c) environment 3 segmentation result

Fig. 10. Experimental results for different simulation environments and segmentation (yellow represents drivable area).

Experiments show that the segmentation accuracy of semantic segmentation network is basically satisfactory. It can be seen that the obstacle information obtained by semantic segmentation is reliable and accurate. The reason why we think the effect is very good is that in the task there are only two types of segmentation. The purpose is to learn the visual features of the intersection between the ground and obstacles (irregular objects, walls, etc.), which is relatively simple compared with general semantic segmentation tasks. In the indoor environment, we only divide the picture into drivable and non-drivable areas. Compared with the complex and changeable outdoor environment, the ground of indoor environment usually has obvious and consistent texture char-

acteristics. In the complete indoor environment, it can also ensure that each column of the image has the intersection of drivable area and non-drivable area.

We also analyze the error caused by semantic segmentation network. Due to the noise of laser sensor data and the semantic segmentation network's own characteristics, segmentation is not completely accurate. The obtained obstacle pixels will have a certain error with the true value. We define 1-pixel error in pixel(x,y) as follows.

$$1-pixel\ error(x,y) = \underset{(m,n)\in(x,y)_{neighbors}}{mean} |d(x,y)-d(m,n)| \tag{10}$$

where (x,y) is the pixel and d is the distance from that obstacle point to robot. The 1-pixel error map obtained under a certain camera pose configuration (h = 40cm, θ = 40°) is shown in the Fig. 10. It can be seen that within 4 meters of field of view, the average 1-pixel error is less than 5 cm. And with the increase of h and θ (camera heading down), the error will decrease. In fact, when the distance to the obstacle is far, we don't need too accurate obstacle location. When the obstacle approaches, the accuracy of semantic segmentation can fully meet the requirement of obstacle avoidance.

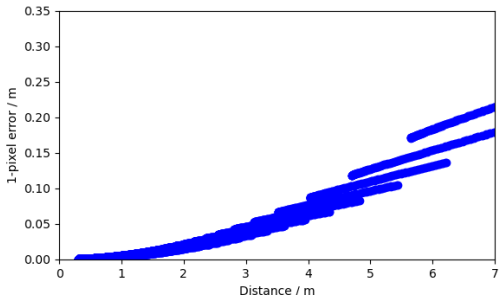


Fig. 11. relationship between 1-pixel error and distance

For the path planning test, set different target points, the robot can basically reach each target point smoothly. When adding obstacles dynamically in the environment, the robot can also plan locally to avoid obstacles. It should be noted that the planning module only takes milliseconds time to solve the formulated optimization problem.

However, the current method still has some shortcomings. For example, the computation of semantic segmentation network is relatively large, so there are certain requirements for robot computing equipment to meet the real-time planning. In addition, due to the limitation of resolution, there is a certain deviation in the position of distant obstacle points, so the length of the planned path cannot be too long.

VI. Conclusion and Future Work

In this paper, We propose a robot navigation framework combining semantic segmentation and path planning to achieve obstacle avoidance, which is vision-based during deployment and low-cost. The most unique feature of our method is that the results of semantic segmentation are transformed into constraints for robot path planning.

Based on the weakly supervised semantic segmentation method in [3], we improved by continuity check and only segment the image into Drivable and non-Drivable regions. We have also done error analysis of segmentation. The simulation experiment shows that this weakly supervised semantic segmentation method is very suitable for indoor environment, and classical semantic segmentation networks such as Segnet can achieve satisfactory segmentation results under appropriate training from 2D-lidar and image data. The segmentation results are then used to generate the boundary conditions required for path planning. This segmentation method does not need to explicitly model environmental obstacles and environmental structures and is universal.

The key obstacle points in the image are mapped to 3D space, and we reduce segmentation noise by establishing probability occupancy map according to the distance information. We also optimize the path planning method. In the path planning stage, a search method is proposed to obtain the boundary constraint, and dynamic weights are set to improve the planning result.

In the future, we plan to deploy the algorithm to real physical mobile robot for testing. Currently it is only designed for static obstacles. Next we will consider dynamic obstacles, and continue to improve the efficiency of the algorithm and improve the real-time performance of the framework.

References

[1] Miyamoto R, Adachi M, Ishida H, et al. Visual navigation based on semantic segmentation using only a monocular camera as an external sensor[J]. Journal of Robotics and Mechatronics, 2020, 32(6): 1137-1153.

[2] Mendez O, Hadfield S, Pugeault N, et al. SeDAR-Semantic Detection and Ranging: Humans can localise without LiDAR, can robots?[J]. 2018.

[3] Barnes D, Maddern W, Posner I. Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy[J]. 2016.

[4] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki and R. Urtasun, "Estimating Drivable Collision-Free Space from Monocular Video," 2015 IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 420-427, doi: 10.1109/WACV.2015.62.

[5] Wurm K M, Stachniss C, Burgard W. Coordinated multi-robot exploration using a segmentation of the environment[C]. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008: 1160-1165.

[6] Guo Y, Liu Y, Georgiou T, et al. A review of semantic segmentation using deep neural networks[J]. International Journal of Multimedia Information Retrieval, 2017.

[7] Liu X, Deng Z. Segmentation of Drivable Road Using Deep Fully Convolutional Residual Network with Pyramid Pooling[J]. Cognitive Computation, 2018.

[8] Wang H, Sun Y, Liu M. Self-Supervised Drivable Area and Road Anomaly Segmentation using RGB-D Data for Robotic Wheelchairs[J]. 2020.

[9] Wang X, Ma H, You S. Deep Clustering for Weakly-Supervised Semantic Segmentation in Autonomous Driving Scenes[J]. Neurocomputing, 2019, 381.

[10] R. Miyamoto, M. Adachi, Y. Nakamura, T. Nakajima, H. Ishida and S. Kobayashi, "Accuracy Improvement of Semantic Segmentation Using Appropriate Datasets for Robot Navigation," 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), 2019, pp. 1610-1615, doi: 10.1109/CoDIT.2019.8820616.
[11] Planning and Decision-Making for Autonomous Vehicles[J]. Annual Review of Control Robotics and Autonomous Systems, 2018, 1(1):annurev-control-060117-105157.
[12] Zhang Y, Sun H, Zhou J, et al. Optimal Vehicle Path Planning Using Quadratic Optimization for Baidu Apollo Open Platform. IEEE. 2020.
[13] Yousuf, S., Kadri, M. Information Fusion of GPS, INS and Odometer Sensors for Improving Localization Accuracy of Mobile Robots in Indoor and Outdoor Applications. Robotica, 39(2), 250-276. 2021.
[14] Mazhar, F., Khan, M.G. Sällberg, B. Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations. Wireless Pers Commun 97, 4467–4491 (2017).
[15] Thrun S, Burgard W, Fox D. Probalistic robotics[J]. Kybernetes, 2006.