Vehicle pose estimation by parking AGV based on RGBD camera

2024 9th International Conference on Automation, Control and Robotics Engineering (CACRE) | 979-8-3503-5030-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CACRE62362.2024.10635074

Xiaopeng Li
Education Center of Experiments and Innovations
Harbin Institute of Technology, Shenzhen
Shenzhen, China
lixiaopeng@hit.edu.cn

Xin Wang
School of Mechanical Engineering and Automation
Harbin Institute of Technology, Shenzhen
Shenzhen, China
wangxinsz@hit.edu.cn

Chen Qu
Research Institute of Bioeconomy
State High-Tech Industrial Innovation Center, Shen Zhen
Shenzhen, China
quc@hiic.com.cn

Jinhua Song
Education Center of Experiments and Innovations
Harbin Institute of Technology, Shenzhen
Shenzhen, China
songjinhua@hit.edu.cn

Hongquan Li
Education Center of Experiments and Innovations
Harbin Institute of Technology, Shenzhen
Shenzhen, China
lihongquan@hit.edu.cn

Yueping Xi
Education Center of Experiments and Innovations
Harbin Institute of Technology, Shenzhen
Shenzhen, China
xiyueping@hit.edu.cn

Abstract*—Parking Automated Guided Vehicles (AGV) continue to play a key role in alleviating urban parking congestion and providing valet parking services. Most of the current parking AGV operate in fixed lanes within stereo garages. The vehicles must be parked in the designated parking area before the parking AGV can complete the vehicle transportation. This places specific demands on the parking driver's skills. When the parking system estimates the vehicle's pose, it typically requires integrating a global camera positioned above the vehicle with sensors mounted on the AGV, including cameras, radars, etc., to achieve the vehicle's pose estimation. This paper proposes a method for the pose estimation of parked vehicles using an RGBD camera installed on a parking AGV. The YOLOv5 network is used to train the model on the RGB images of license plates and wheels, respectively. Subsequently, the generated model file is deployed to the parking AGV. YOLOv5 license plate recognition model detects license plates. Through coordinate conversion, conditional filtering, and point cloud registration, the pose transformation matrix of the vehicle is obtained, and the horizontal rotation angle of the parking AGV is derived. The YOLOv5 wheel recognition model retrieves the bounding box of wheel detection information. By utilizing wheel symmetry, the Euclidean distance between the front and rear wheel coordinate points is computed to ascertain the vehicle's wheelbase and the distance between the parking AGV and the wheels. Subsequently, the distance to park the AGV is calculated. Moving distance. The sensors utilized in this method are solely RGBD cameras, which decrease the number of necessary sensors, improve the maneuverability of the parking AGV, and eliminate track constraints. The proposed method reduces the cost of parking AGV pose estimation and simplifies the detection steps, thereby enhancing detection speed and efficiency.

Keywords—parking AGV, vehicle pose estimation, YOLOv5, RGBD camera

I. INTRODUCTION

The advancement of industrial technology and the empowerment of artificial intelligence (AI) have made industrial robots more intelligent and efficient. In the field of valet parking[1], there are currently two common solutions. One system is based on a sophisticated parking infrastructure, such as a stereo garage[2-4], utilizing parking Automated Guided Vehicles (AGV)[5], parking areas, and a control system to offer valet parking services. The other system is an intelligent system integrated within the vehicle itself, enabling valet parking functionality through control algorithms and sensor data fusion[6]. At present, the market still has a relatively low occupancy rate of vehicles equipped with intelligent systems, and stereo garages remain the primary method for alleviating parking pressure. In a stereo garage, a closed parking area is necessary for users to park their vehicles and for AGV to transport vehicles within the garage. During the process of AGV picking up the car, the parking area will be closed. This period of time will cause users queuing for parking to continue waiting, which will greatly affect the user experience. At the same time, because the stereo garage requires a dedicated parking area, it cannot be adapted from the current garage, leading to increased construction expenses.

In order to reduce construction costs and to upgrade the stereo garage based on the existing garage, it is essential to improve the working freedom of the parking AGV. In existing stereo garages, the parking AGVs are generally based on fixed tracks to complete vehicle handling work. If the parking AGV is empowered with AI so that it can autonomously identify the vehicle's position and transport it, then there will be no need to build tracks, which can save costs to a great extent. It can be seen that it is important for the parking AGV to accurately identify the vehicle's position and enter directly under the vehicle to complete the lifting and transportation work.

In the field of vehicle pose recognition, a common method is to process point cloud data from laser sensors. Bo Gu[7] fuses lidar point cloud and road features for vehicle pose estimation. Jing Sun[8] completes vehicle pose estimation by fitting and filtering radar point clouds. Ningning Ding[9] used radar point cloud to create an Efficient Convex Hull-Based method to identify vehicle pose. The symmetry of the wheel is the most commonly used reference basis in the process of vehicle pose recognition and solution. Zhixiong Ning[10] used the point cloud and the symmetry of the wheel to complete the solution of the vehicle pose. Guohong Yi[11] builds a wheel interpolation model through point clouds and uses wheel symmetry to improve the robustness of attitude estimation. In addition to lidar, there are also other types of sensors, such as using pressure sensors to achieve pose estimation. Juzhong Zhang[12] uses a pressure sensor array combined with rectangular fitting using the least squares method to improve accuracy. In addition to pose estimation, this form can also obtain basic parameters of the vehicle. Due to the price of multi-line lidar, the cost of parking AGV has increased, causing difficulties in the process of application popularization.

As the target recognition algorithm matures, it has been widely used in industrial applications, such as in human-robot interaction, industrial robotic arms, unmanned AGV and other fields[13-14]. The process of identifying and transporting vehicles by the parking AGV is similar to the process of identifying and lifting pallets by a forklift. The method used by forklifts to identify and lift pallets is through depth cameras. Chunghyup Mok [10] and Efthimios Tsiogas[11] use the depth camera to complete the unique feature recognition of the target through deep learning, and complete the alignment and lifting of the pallet through the distance detection of the depth camera. This method provides an idea for parking AGV to identify vehicle poses through depth cameras.

In this study, a vehicle pose recognition method based on RGBD depth cameras is proposed. This method offers more degrees of freedom for the movement of parking AGVs. Additionally, utilizing only depth camera sensors reduces production costs and retrofit costs. The solution addresses the issue of stereo garage parking AGVs having to depend on tracks to retrieve cars, which consumes the user's time. It provides a new application method for a future stereo garage system that is more intelligent, convenient, and easy to modify.

The structure of this paper is as follows: Section 2 introduces the overall process of the proposed vehicle pose estimation. Section 3 details the implementation process of the pose estimation algorithm based on YOLOv5. Section 4 describes the experiment and results. Finally, we summarized the overall of this paper in Section 5.

II. PROPOSED VEHICLE POSE ESTIMATION ARCHITECTURE

The proposed vehicle pose estimation architecture consists of two subsystems, namely the parking vehicle image real-time processing system and the parking AGV pose adjustment system, as shown in Fig. 1.

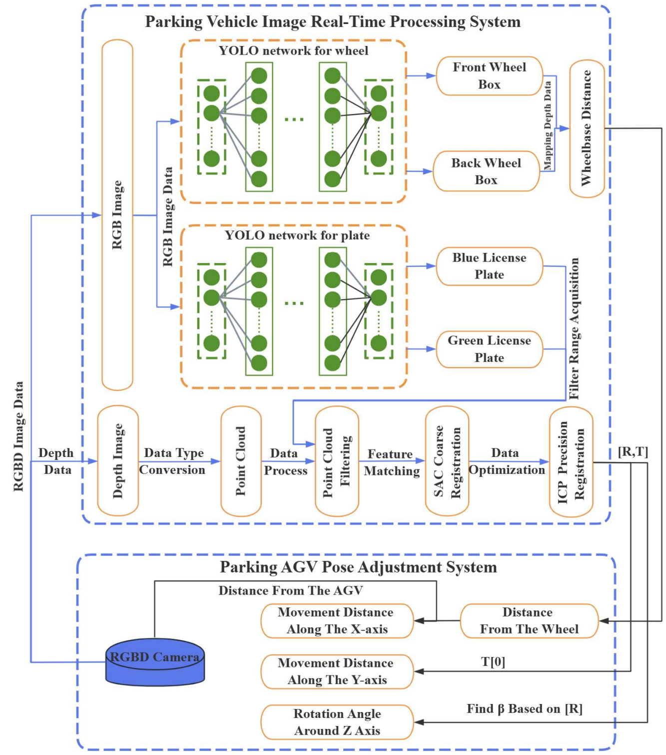


Fig. 1. The proposed vehicle pose estimation architecture

A. *The parking vehicle image real-time processing system*

The parking vehicle image real-time processing system divides the RGBD depth image obtained from the parking AGV into two parts for processing. One part involves RGB image processing, utilizing the YOLOV5 training model for target recognition and coordinate acquisition, while the other part focuses on depth image processing. After converting the depth image into a point cloud image, the rotation and translation matrix of the parking AGV is obtained through license plate point cloud feature extraction and matching.

B. *The parking AGV pose adjustment system*

The parking AGV pose adjustment system involves the output of RGBD camera data and the input of pose solution results. After the parking AGV reaches the designated location, the equipped RGBD camera continuously captures image information and transmits the data to the parking vehicle real-time image processing system. After image processing, the calculated movement distance and rotation angle are sent back to the parking AGV. Subsequently, the parking AGV adjusts its position and orientation to achieve alignment with the parked vehicle.

III. METHODS OF VEHICLE POSE ESTIMATION

This section specifically explains how to use RGBD cameras to achieve vehicle pose estimation and adjust the pose of parking AGV. In Part III-A, the dataset collection method for target recognition is introduced. Part III-B demonstrates the process of model training using YOLOv5, and Part III-C outlines the method of establishing a reference template before solving the pose. Part III-D discusses the specific process of pose solution.

A. Datasets collection

The license plate dataset is sourced from the Chinese City Parking Dataset (CCPD), which includes CCPD2019 and CCPD2020. CCPD2019 is a dataset of blue license plates for gasoline vehicles, while CCPD2020 consists of green license plates for new energy vehicles. The license plate dataset covers various situations in different environments, weather conditions, locations, and angles. 2,500 images from each of the two CCPD datasets formed the dataset for this training. In each image, there is a corresponding XML file that stores the license plate category and bounding box information. Convert XML format tags to YOLO format tag sets and randomly split them into training and test sets at a 9:1 ratio.

The wheel dataset was collected in the garage using an RGB camera. The RGBD camera of choice is the RealSense D455, which is a stereoscopic depth camera with a global shutter RGB sensor and IMU. The D455 camera can extend the distance between depth sensors to 95 mm, improving depth error to less than 2% within 4 m. The settings used in the camera sensor during image acquisition were as follows: VGA resolution (640x480) and 30 frames per second. Install the RGBD camera on the parking AGV or approximately 10 cm above the ground. A total of 5,000 images of cars at various distances and angles in different lighting conditions were collected for different types of cars parked in parking lots. Using the open-source annotation tool LabelMe to annotate all RGB images, of which over 15,000 front and rear wheels. Each image had a separate annotation file formatted as a JSON file, which cannot be directly imported into the YOLOv5 model. Convert the annotated JSON format labels to YOLO format and split the image dataset and its corresponding annotation dataset into a training set and a test set at a ratio of 9:1 to prepare for subsequent model training. The Wheel dataset contains 4,500 training and 500 test images, respectively.

B. YOLOv5 model training

Object detection is a computer vision task that involves distinguishing the target of interest from other areas in an image or video and providing the current location of the target in the image by marking it with a bounding box. With the advancement of deep learning, target detection based on deep learning has become widely utilized due to its outstanding performance. At present, classic target detection methods mainly include single-stage methods, such as YOLO, SSD, RetinaNet, and key-based detection methods, as well as multi-stage methods like Fast R-CNN, Faster R-CNN, and Cascade R-CNN. Since the characteristics of the target to be detected are relatively distinct and somewhat different from the environment, the number of dataset does not need to be too large. Therefore, the YOLOv5 algorithm was selected for model training. It enables predictions for object localization and classification simultaneously. Each image was accompanied by a separate annotation file formatted as a JSON file, which cannot be directly imported into the YOLOv5 model. Convert the annotated JSON format labels to YOLO format and split the image dataset and its corresponding annotation dataset into a training set and a test set at a 9:1 ratio to prepare for subsequent model training. The Wheel dataset contains 4,500 training and 500 test images, respectively.

This study utilizes the Google Colaboratory platform, which is a deep learning framework that provides free GPU computing capabilities. The platform's environment configuration includes PyTorch 2.1.0 , CUDA 12.1, CUDNN 8906 and Pillow 8.3.2. For object detection, the weights are initialized with pre-trained YOLOv5s, and the batch size is set to 16. Key hyperparameters include an initial learning rate schedule of 0.0032, a momentum of 0.779, a warmup_epochs of 1.33, a warmup_momentum of 0.86, and a weight decay of 0.00058. Import the license plate label set and wheel label set to train and test the pretrained YOLOv5s model. Assign the output result to the image coordinates of the target bounding box. The coordinates will be used for target reprocessing. The main average precision (mAP) was used as the evaluation metric. After training, two model files are obtained, named the wheel recognition model and the license plate recognition model. For each detection, the model detector outputs coordinate of the target and a confidence score. Deploy the model file on the AI edge computing device of the parking AGV using Fast Deep Learning Model Deployment (FastDeploy).

C. Preparation of templates

Since there are many types of vehicles with different sizes and parameters, the pose solution needs to rely on a standard size or template. Wheels and license plates are good choices as templates. The pose of the wheels depends on the driver's control of the steering, making it unsuitable as a template. Since the size of the license plate is standardized according to national regulations, the dimensions of the gasoline blue license plate are 440mm x 140mm, while the new energy green license plate measures 480mm x 140mm. Therefore, the license plate can be used as a standard size and reference object in vehicle pose estimation methods. Although vehicles come in various shapes and sizes, the current location of a vehicle can be determined by the license plate, which is typically mounted in the middle of the vehicle. Therefore, it is necessary to collect license plate information as a template for implementing the vehicle pose estimation method.

First, park the vehicle in the middle of the parking space. Move the parking AGV 0.6 meters in front of the vehicle and position it directly facing the vehicle. The D455 RGBD camera mounted on the parking AGV captures and stores RGB images and depth images of the vehicle. Use the license plate recognition model trained by YOLOv5 to identify the saved RGB image and extract the 2D coordinates of the license plate bounding box. The depth image is converted into a point cloud image. The coordinates of the bounding box are used to define the point cloud filtering conditions, enabling the extraction of the license plate point cloud information, which is then utilized as a reference template. The 3D coordinates are obtained by converting coordinates between the RGB image sensor and the depth sensor of the camera. The specific acquisition method is as follows:

$$\boldsymbol{x_i} = [x_i \; y_i \; z_i]^{\mathrm{T}} = \begin{bmatrix} z_i \cdot (u_i - C_x)/(F_x \cdot depScale) \\ z_i \cdot (v_i - C_y)/(F_y \cdot depScale) \\ z_i \end{bmatrix} \quad (1)$$

where (u_i, v_i) are the 2D coordinates of point i in the image coordinate system, and $\boldsymbol{x_i}$ is the 3D coordinates corresponding to point $\boldsymbol{i}$ in the camera coordinate system. C_x, $C_y$represent the offsets of the camera's optical axis on the x-axis and y-axis of the image coordinate system, while $F_x$and F_y denote the zoom focal length of the camera. $depScale$ is a parameter that converts pixels to meters.

Second, use the method from the first step to create two templates: one for a new energy green license plate and one for a gasoline blue license plate. These templates will serve as reference bases for point cloud registration to derive rotation and translation matrices.

D. Pose adjustment of parking AGV

The pose adjustment of the parking AGV is divided into two parts: one part involves the horizontal rotation angle, and the other part involves the movement distance. The horizontal rotation angle can be determined by aligning the license plate of the parked vehicle in the parking area with the template vehicle point cloud information and calculating it using the resulting rotation matrix. The distance traveled is calculated based on the vehicle's wheelbase determined through wheel recognition, the distance between the wheel and the camera, and the longitudinal distance of the camera from the center of the AGV. The schematic diagram in Fig. 2 illustrates the adjustment of the parking AGV's pose.

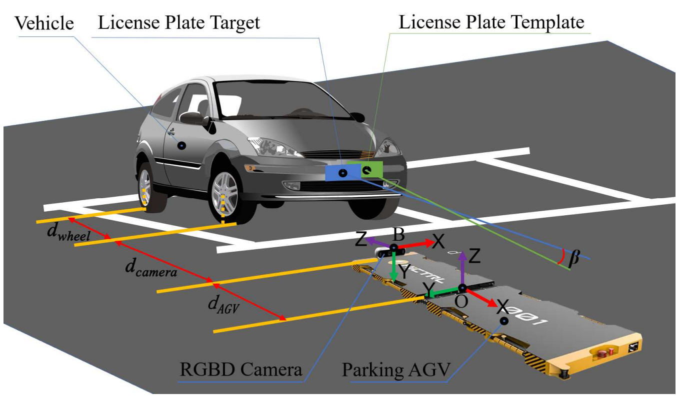


Fig. 2. The schematic diagram of parking AGV pose adjustment

First, figure out the parked AGV's horizontal rotation angle. Place the parking AGV where the template for the license plate is made. It's not necessary to be very precise about this location. Utilize the license plate recognition model to finalize the recognition of the license plate. Convert the two-dimensional coordinates of the license plate into the 3D coordinate system within the camera coordinate system to determine the upper left corner point $(\boldsymbol{x_1})$ and lower right corner point $(\boldsymbol{x_2})$ of the license plate bounding box. To configure point cloud filtering settings and obtain the 3D point cloud of the license plate, use its 3D coordinates.

$$\boldsymbol{x_1} = [x_1\ y_1\ z_1]^{\mathrm{T}} \quad \boldsymbol{x_2} = [x_2\ y_2\ z_2]^{\mathrm{T}} \tag{2}$$

$$\begin{cases} x_{min} = x_1, & x_{max} = x_2 \quad if\ x_1 < x_2 \\ x_{min} = x_2, & x_{max} = x_1 \quad else \end{cases}$$

$$\begin{cases} y_{min} = y_1, & y_{max} = y_2 \quad if\ y_1 < y_2 \\ y_{min} = y_2, & y_{max} = y_1 \quad else \end{cases} \tag{3}$$

$$\begin{cases} z_{min} = z_1, & z_{max} = z_2 \quad if\ z_1 < z_2 \\ z_{min} = z_2, & z_{max} = z_1 \quad else \end{cases}$$

$$\mathbf{P}_{\boldsymbol{t}} = \left\{ \left[x_j, y_j, z_j\right]^{\mathrm{T}} \right\} = \left\{ \begin{bmatrix} x_{min} \le x_j \le x_{max} \\ y_{min} \le y_j \le y_{max} \\ z_{min} \le z_j \le z_{max} \end{bmatrix} \right\} \tag{4}$$

where $\mathbf{P}_{\boldsymbol{t}}$ represents a range of target license plate point clouds.

The actual collected license plate point cloud and the template point cloud are aligned using Sample Consensus Initial Alignment (SAC-IA) method and the Iterative Closest Point (ICP) method to derive the rotation and translation transformation matrix:

$$\mathbf{R}, \mathbf{t} = \mathrm{argmin}_{R,t} \frac{1}{|\mathbf{P}_S|} \sum_{i=1}^{|\mathbf{P}_S|} \left\| \mathbf{P}_t^i - (\mathbf{R} \cdot \mathbf{P}_S^i + \mathbf{t}) \right\|^2 \tag{5}$$

where $\mathbf{P}_s$ represents a range of template point clouds, $\mathbf{R}$ is the transformation matrix of the template point cloud relative to the target point cloud, and $\mathbf{t}$ is displacement between the two. The angles of rotation around the Z, Y, and X axes of the camera coordinate system are denoted by γ, β, and α, respectively. The formula for the horizontal rotation angle (β) of the parking AGV and its movement (d_y) along the Y-axis is:

$$\mathbf{R} = \mathbf{R}_{\boldsymbol{z}}(\gamma) * \mathbf{R}_{\boldsymbol{y}}(\beta) * \mathbf{R}_{\boldsymbol{x}}(\alpha) \tag{6}$$

$$\begin{cases} \mathbf{R}_{\boldsymbol{x}}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \\ \mathbf{R}_{\boldsymbol{y}}(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \\ \mathbf{R}_{\boldsymbol{z}}(\gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{cases} \tag{7}$$

$$\mathbf{t} = [t_1 \quad t_2 \quad t_3]^{\mathrm{T}} \tag{8}$$

$$d_y = t_1 \tag{9}$$

Second, determine the parked AGV's movement distance along the X-axis.

Before calculating the AGV movement distance, it is essential to assess if the parking AGV can enter under the vehicle chassis to prevent any damage to the chassis. The height of the bounding box of the front wheel is used as the chassis height, and the distance between the two wheels is used as the chassis width. The parking AGV will determine whether it can enter based on the size of the chassis to prevent scratches. The 3D coordinates of the points $(\boldsymbol{x_i})$ on the front wheel bounding box in the camera coordinate system can be determined using the wheel recognition model and the coordinate mapping relationship. The points $(\boldsymbol{x_i})$ of wheel bounding box distribution are shown in Fig. 3 Chassis dimensions can be derived:

$$\boldsymbol{x_i} = [x_i\ y_i\ z_i]^{\mathrm{T}} \tag{10}$$

$$d_h = (y_3 - y_4 + y_5 - y_6)/2 \tag{11}$$

$$d_w = x_8 - x_7 \tag{12}$$

where d_h and d_w respectively represent the height and width of the chassis.

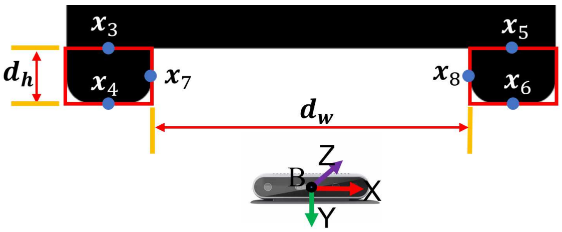


Fig. 3. The distribution points of wheel bounding box

To ensure that the parking AGV can enter directly under the vehicle for transportation to the stereo garage, acquiring the wheelbase is critical for the AGV to perform the vehicle handling tasks. After the parking AGV completes the pose adjustment based on the rotation angle calculated from the license plate registration, the front wheels and rear wheels are detected respectively through the wheel recognition model. Given coordinate limitations, position the front and rear wheels on a single side based on wheel symmetry. Through coordinate transformation, the 3D coordinates of the two points in the depth camera are obtained from the two-dimensional coordinates of the bottom center of the front and rear wheel bounding boxes, denoted as $\boldsymbol{x_4}$ and $\boldsymbol{x_9}$, respectively. The wheelbase (d_{wheel}) can be found by measuring the separation between two points:

$$\boldsymbol{x_4} = [x_4\ y_4\ z_4]^{\mathrm{T}} \quad \boldsymbol{x_9} = [x_9\ y_9\ z_9]^{\mathrm{T}} \tag{13}$$

$$d_{wheel} = \sqrt{(x_9 - x_4)^2 + (y_9 - y_4)^2 + (z_9 - z_4)^2} \tag{14}$$

$$d_{camera} = z_4 \tag{15}$$

Keep the parking AGV stationary, continuously collect images 50 times, and calculate the wheelbase. Then remove the maximum and minimum values of the wheelbase, and calculate the average of remained data. The movement distance d_x of the parking AGV can be derived:

$$d_x = d_{wheel} + d_{camera} + d_{AGV} \tag{16}$$

where d_{AGV} is the longitudinal distance between the camera and the AGV center, which can be obtained by measurement.

IV. Experiments and Results

A. Performance of license plate detection

The number of epochs is adjusted until the loss function converged. It takes approximately 60 minutes to train a model over 200 epochs. The model converges quickly in the first 100 epochs and almost converges at 200 epochs. Then an early stopping mechanism is applied to select the optimal weights.

There are hardly any objects resembling the targets due to the stable environment around the parking AGV, which has a limited variety of targets. Consequently, to meet the recognition requirements, the probability of identifying the target must be greater than 0.8. The best-performing license plate recognition model configuration has a precision of 0.963, a recall of 0.983, a mean average precision (IOU:0.5) of 0.988, and a mean average precision (IOU:0.5-0.95) of 0.867. The license plate recognition model can meet the requirements.

To better evaluate the model, images taken at various distances and angles under different lighting conditions are selected for further model inference detection. Then the image detection inference is carried out, as shown in Fig.4. The model detector can accurately classify and locate license plates, including those of oil and new energy vehicles.

B. Template of license plate

Object detection was developed using a 2D detector operating on RGB images, where each 2D bounding box defines a 3D frustum region. The point cloud of the vehicle can be obtained using the PCL library, as shown in Fig. 5.a. The point cloud of a license plate can be filtered based on 3D coordinates, as illustrated in Fig. 5.b.

C. Wheel detection and wheelbase calculation

Fig. 6 illustrates the improvement of model by displaying various performance metrics for the training and validation sets. The model improved swiftly in terms of recall, precision and mean average precision before plateauing after about 300 epochs.

The optimal weight was then used for deference. As shown in Fig. 7, in the example featuring various types of vehicles parked in the parking lot, YOLOv5s can accurately classify most of the wheels, including front and rear wheels. Compared with the detection results for the license plate dataset, the detection performance for the wheel dataset has significantly decreased. This indicates that the detection difficulty of the wheel dataset is higher than that of the license plate dataset.

In Fig. 7, the vehicle type and actual wheelbase are shown in the first row at the bottom of the picture, while the wheelbase value calculated by the algorithm is displayed in the second row. After experimental testing, the error between the measured wheelbase value and the actual value is within 3cm. This error can be offset by the AGV's self-regulation. Experiments show that by integrating wheelbase detection with parking AGV self-regulation, the parking AGV can move directly under the vehicle chassis and lift the vehicle.


Fig. 4. Performance of the license plates detection

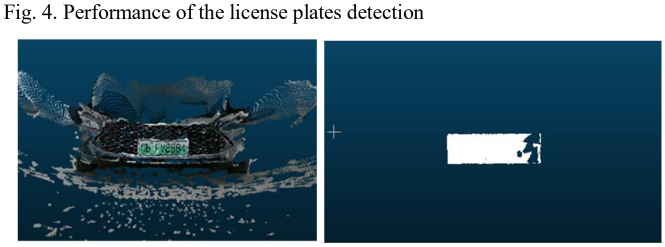

(a) Point cloud of vehicle (b) Point cloud of license plate

Fig. 5. The point cloud template of license plate

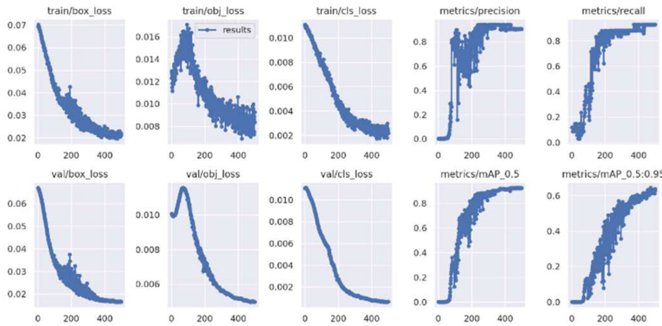


Fig. 6 Performance of the wheel detection model

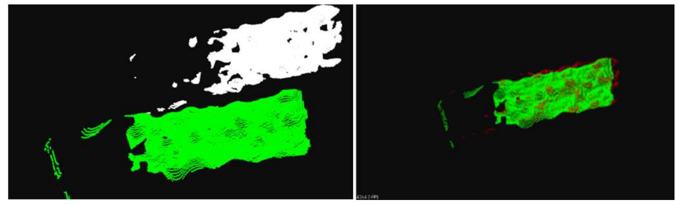
Fig. 7. Performance of the wheel detection and wheelbase calculation

D. *Vehicle pose estimation*

Acquiring the AGV's moving angle is just as important as figuring out its movement distance. The rotation and translation transformation matrices of the target license plate point cloud and the template license plate point cloud were calculated in 0.18 seconds. In Figure 8, the white point cloud represents the target point cloud, the green point cloud represents the template point cloud, and the red point cloud is obtained by multiplying the target point cloud by the transformation matrix. This study conducts coarse and fine registration of the point cloud data extracted through conditional filtering and the template data. As depicted in Fig. 8.b, the overlap of the registered point cloud is substantial, indicating precise pose estimation. Upon establishing the horizontal rotation angle and factoring in the displacement, the parking AGV can effectively position itself at the center of the vehicle's base.

V. CONCLUSION

The proposed method can achieve vehicle pose estimation and parking AGV motion adjustment using only RGBD cameras, enabling parking AGV in stereo garages to no longer rely on guide rails. The application of the YOLOv5 model enhances the intelligence of the parking AGV, making it suitable not only in stereo garages but also in ordinary garages. Template creation only needs to be done once, which reduces a significant amount of work time and operational steps for preparation. The detection process involves only two steps: the first step is to obtain the horizontal rotation angle, and the second step is to determine the moving distance. The proposed method improves the operation efficiency and reduces costs by minimizing the use of sensors.



(a) The target and template license plate point cloud before registration

(b) The target and template license plate point cloud after registration

Fig. 8. Before and after registration of license plate point cloud

ACKNOWLEDGMENT

The work is supported by the Shenzhen Science and Technology Program (JSGG20211029095205007 and JCYJ20210324115812034), the Shenzhen Science and Technology Major Program (KJZD20231023100459001).

REFERENCES

[1] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad, and M. K. Khan, “From smart parking towards autonomous valet parking: A survey, challenges and future Works,” *Journal of Network and Computer Applications,* vol. 175, 2021.

[2] W. Wang, S. Qin, B. Chen, and Y. Ge, “Design of Intelligent Parking Lot System Based on Dynamic Priority Strategy,” in 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE), 2021, pp. 241-245.

[3] X.-f. Huang, W. Zhang, and Y.-m. Wang, “Intelligent Optimization Design of A Simple Mechanical Stereo Garage,” in 2022 3rd International Conference on Intelligent Design (ICID), 2022, pp. 218-223.

[4] J. Huang, Q. Chen, and F. Xie, “Design of Intelligent Stereo Garage System Based on PLC Control,” Procedia Computer Science, vol. 208, pp. 88-93, 2022.

[5] Q. Q. Kai Shen, Qiaoling Wu, Zhijian Lin and Yeqing Wu*, “Research on the development status of AGV parking robot,” *Journal of Physics: Conference Series,* vol. 1905, no. 1, 2021.

[6] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, “Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver,” IEEE Transactions on Automation Science and Engineering, 2022.

[7] B. Gu, J. Liu, H. Xiong, T. Li, and Y. Pan, “ECPC-ICP: A 6D vehicle pose estimation method by fusing the roadside lidar point cloud and road feature,” Sensors, vol. 21, no. 10, pp. 3489, 2021.

[8] J. Sun, Y.-M. Ji, and S.-D. Liu, “Dynamic Vehicle Pose Estimation with Heuristic L-Shape Fitting and Grid-Based Particle Filter,” Electronics, vol. 12, no. 8, pp. 1903, 2023.

[9] N. Ding, “An Efficient Convex Hull-based Vehicle Pose Estimation Method for 3D LiDAR,” arXiv preprint arXiv:2302.01034, 2023.

[10] Z. Ning, X. Wang, J. Wang, and H. Wen, “Vehicle pose estimation algorithm for parking automated guided vehicle,” International Journal of Advanced Robotic Systems, vol. 17, no. 1, pp. 1729881419891335, 2020.

[11] G. Yi, Z. Cao, X. Xu, B. Wu, and J. Li, "Real-time Vehicle Pose Estimation Based on Nonlinear Optimization and Probability Interpolation." pp. 99-105.

[12] J. Zhang, Y. Chu, Z. Wang, T. Ye, L. Cai, and H. Yang, "Vehicle Pose Estimation System Base on Pressure Sensor Array for Clamping Parking Robot." pp. 95-100.

[13] Y. Luo, S. Li, and D. Li, “Intelligent perception system of robot visual servo for complex industrial environment,” Sensors, vol. 20, no. 24, pp. 7121, 2020.

[14] Y. Y. Liau, and K. Ryu, “Status recognition using pre-trained YOLOv5 for sustainable human-robot collaboration (HRC) system in mold assembly,” Sustainability, vol. 13, no. 21, pp. 12044, 2021.