



# A Real-Time Detection Algorithm for Semi-structured Boundaries in the Park Based on 3D LiDAR

Chao Ban, Ruijuan Chi<sup>(✉)</sup>, Tong Su, Yueqi Ma, and Xiulian Huang

College of Engineering, China Agricultural University, Beijing, China  
chiruijuan@cau.edu.cn

**Abstract.** Efficient and accurate real-time detection of semi-structured driving boundaries within the park is required by unmanned ground vehicles (UGVs) in order to facilitate vehicle positioning and path planning. These boundaries have a regular shape, but do not have obvious line and surface features, such as low shrubs. It is crucial for ensuring optimal navigation for UGVs. We have developed a new method for detecting semi-structured boundaries based on 3-dimensional Light Detection and Ranging (3D LiDAR), which involves three key steps. The first step involved pre-processing the 3D raw laser point cloud, which included three tasks: coordinate correction, segmenting the region of interest (ROI), and ground point cloud segmentation to achieve point cloud down-sampling. Next, the non-ground points in ROI were projected onto the ground, and boundary feature points that affect the driving of UGV were extracted. To eliminate incorrect boundary points, a two-step method consisting of the Random Sample Consensus (RANSAC) algorithm and inter-frame boundary line angle elimination was used. In the third step, the data of Position and Orientation System (POS) were integrated to construct a global point cloud map of the semi-structured driving boundary. An experiment was conducted on a self-developed UGV-LiDAR platform to evaluate the performance of the proposed algorithm. It was indicated by the results of the study that the feature point detection rate was exceeded by 98.95%, and the average processing time per frame was around 45.7 ms when there are more shrubs, meeting the real-time demands. This research provided valuable insights into perception technology for security operations of UGVs in parks.

**Keywords:** real-time · driving boundary detection · UGV · LiDAR · park

## 1 Introduction

In recent years, the use of unmanned ground vehicles (UGVs) has become increasingly popular in parks for tasks such as transporting items, conducting safety patrols, and disposing of garbage [1, 2]. However, in order to operate safely and autonomously in such an environment, effective environmental perception technology is required for UGVs,

---

This work was supported by the National Natural Science Foundation of China under Grant 52172396.

with driving boundary detection and mapping being a crucial component. This technology provided a basis of path planning and decision-making [3–5]. While structured boundaries, such as curbs and walls, had been extensively studied as important components in forming driving boundaries, common low shrubs were also significant as they became boundaries of the driving area and limited the movement of UGVs. Moreover, in situations where the Position and Orientation System (POS) signal was obstructed by tall buildings and trees, the driving boundary became an effective feature to guide the UGVs [6].

Extensive research has been conducted over the past few decades on computer vision-based methods, which are known for their low cost and ability to extract rich texture features. These methods mainly rely on visual sensors, such as monocular and stereo cameras, to identify and segment boundaries based on color transformations between driving and non-driving areas in the environment [7–9]. Panev et al. [7] proposed a method for accurately estimating the 3D parameters of structured curbs using a monocular camera. Their approach combined 3D geometric reasoning with advanced visual-based detection methods to estimate real-time distance from vehicles to the curb, as well as the direction, height, and depth of the curb. Cheng et al. [8] proposed a disparity map generated using stereo vision matching to estimate flat areas on the road surface, and proposed an effective 16-dimensional descriptor based on the appearance, geometric shape, and disparity features of curbs to extract curb boundary. However, cameras were highly sensitive to changes in light intensity, and as a result, their detection accuracy was significantly impacted when faced with low-light conditions. This limitation rendered them ineffective for use during nighttime surveillance.

When compared to cameras, Light Detection and Ranging (LiDAR) is an active detection sensor that is relatively immune to the effects of ambient light, making it a viable option for nighttime operations. According to the number of laser harnesses, it could be divided into two types: 2D LiDAR and 3D LiDAR. The 2D LiDAR, with its single wire harness, boasted a lower cost and has been applied in various fields. Demir et al. [10] introduced an adaptive approach to extract road boundaries by utilizing 2D LiDAR sensors in their study. The proposed method involved a three-stage detection algorithm that allowed for the adaptive updating of the dataset. The 2D LiDAR scanned a plane at a fixed elevation angle and captures a range of depth and angle measurements for each scan. Wijesoma et al. [11] have introduced a novel approach that employed extended Kalman filtering and continuous distance/azimuth readings from a 2D LiDAR measurement system for swift detection and tracking of curb. However, due to the sparsity of points provided by 2D LiDAR per frame, only a limited number of points were available on the boundary. This made it challenging to extract useful features from 2D LiDAR point clouds, particularly for semi-structured shrub boundaries.

Compared to 2D LiDAR, 3D LiDAR offers superior performance in terms of accuracy, resolution, recognition speed, and error rate. However, its high cost remained a barrier to its widespread adoption in UGVs [12, 13]. One of the most commonly used methods was to establish a 2D grid map based on a large amount of point cloud data, and determine local geometric features based on the elevation changes within each grid to detect UGV driving boundary. Huang et al. [14] proposed a technique for detecting curbs by utilizing global road trends and extracting update mechanisms. This involved

projecting the segmented ground point cloud and estimating the position and height of curbs based on the road centerline and width. On the other hand, Borja et al. [15] rasterized the 3D point cloud to make it more manageable and reduce its size. They set threshold values based on curb boundaries and binarized the grid image to detect boundary feature points.

A novel approach for detecting semi-structured driving boundaries using 3D LiDAR in park environments has been proposed. The approach integrated position and attitude to construct a global point cloud map that provided accurate boundary information for UGVs driving in the park. The method was capable of detecting low shrubs and other semi-structured objects that may pose a challenge to traditional methods. The real-time detection capability of the method made it ideal for use in dynamic park environments. This paper is organized as follows: Sect. 2 provides an overview of the UGV-LiDAR system utilized in this study; Sect. 3 describes the in-depth explanation of the proposed method for identifying semi-structured boundaries; Sect. 4 shows the results of boundary detection and mapping conducted within the park; Sect. 5 presents our conclusions and future research endeavors.

## 2 UGV-LiDAR System Development

### 2.1 Hardware Information

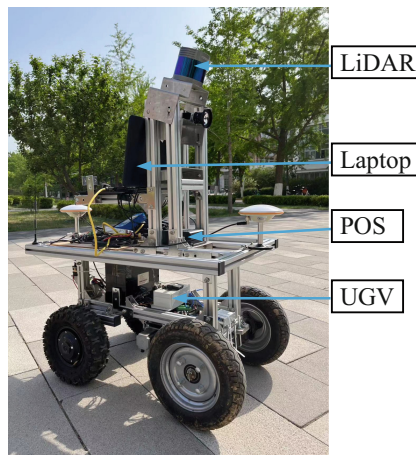
The UGV used in this study was a self-developed wheeled platform for park inspection, which equipped with a 3D LiDAR sensor for environment perception and a POS for accurate localization. The platform was built with robust materials and had a low center of gravity, making it stable and capable of traversing uneven terrain. The LiDAR employed on the UGV was RS-LiDAR-32 (RoboSense, China), which integrated 32 laser transceiver components and rapidly rotated while emitting high-frequency lasers to continuously scan the external environment. It utilized a hybrid solid-state LiDAR technology that recorded distance and reflectivity information through the 32 lasers, which operated at a wavelength of 905 nm. Depending on the rotation speed and return mode, the LiDAR could take up to 578,688 laser measurements per second with a  $\pm 3$  cm range accuracy within a distance of 200 m, and had a vertical/horizontal field of view (FOV) of  $40^\circ/360^\circ$ . The POS occupied by the UGV was CGI-610(CHC, China), which combined Global Navigation Satellite System (GNSS) with Inertial Measuring Unit (IMU) using data fusion technology. The CGI-610 adopted a full system multi-frequency scheme in satellite positioning, receiving signals from multiple frequency bands such as GPS, GLO, BDS, and Galileo. In response to the situation where trees and buildings in the park were prone to blocking satellite signals, the IMU was equipped with high-precision gyroscopes and accelerometers. With the help of the new generation of multi-sensor data fusion technology, high-precision carrier position, attitude, speed, and sensor information were provided in real-time, providing support for long-term operation of UGV in complex environments. It could output 100 Hz high-frequency data, guiding UGVs to travel within the park with  $\pm 0.02/0.03$ m horizontal/vertical positioning accuracy via real-time kinematic technology. A high-performance laptop was equipped to collect and process data. The laptop was equipped with an i5-7300 CPU, 8 GB of RAM,

and a 256 GB SSD for both Ubuntu20.04 operating system and data storage. Table 1 summarized the main hardware technical specifications.

**Table 1.** The Main Technical Parameters of UGV-LiDAR System Components.

Hardware	Main parameters
RS-LiDAR-32	32 lasers with a 905 nm wavelength, $40^\circ \times 360^\circ$ FOV, $0.1^\circ/0.2^\circ/0.4^\circ$ horizontal resolution, 200 m measurement range, $\pm 3$ cm range accuracy, max 578,688 measurements per second
CGI-610	GNSS: $\pm 0.03$ m vertical and $\pm 0.02$ m horizontal accuracy via real-time kinematic IMU: bias stability of Gyroscope with $2.5^\circ/\text{h}$ , bias stability of Accelerometer with $3.6 \mu\text{g}$
Laptop	i5-7300 CPU, 8 GB RAM, 256 GB storage, fast Ethernet

The UGV-LiDAR system was equipped with a flexible aluminium profile bracket that allowed for easy installation and placement of the LiDAR, POS, and laptop. The LiDAR was strategically positioned on top of the UGV, ensuring that its center was as close as possible to the geometric center of the IMU sensor at the POS. Meanwhile, the laptop was placed behind the pillar to avoid obstructing the POS signal and LiDAR's FOV. The UGV-LiDAR system used in this study was shown in Fig. 1.

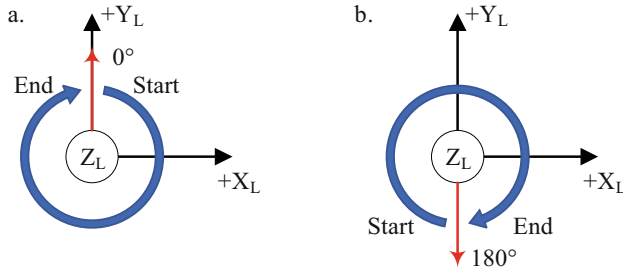


**Fig. 1.** The UGV-LiDAR System.

## 2.2 LiDAR Configuration

The UGV targeted in this study had a low speed of 0.5m/s and could only be moved in a front wheel steering to rear wheel drive manner, without complex motion conditions. Thus, the vertical angular resolution of the laser transmitter was uneven, and the downward tilt of  $15^\circ$  could make the high-density area of the point cloud closer to the UGV and improved the near perception ability. If the laser transmitter was installed at a larger angle, it would scan the UGV itself and wasted the perception ability. Considering the point cloud transmission bandwidth and angular resolution, the LiDAR rotation speed was set to 600 rpm with a horizontal angular resolution of  $0.2^\circ$ . However, the complexity of the interlaced leaves of shrubs could result in multiple reflections when a laser pulse was emitted, due to the divergent nature of the beam. As a general rule, the receivers' ability to detect targets weakens as the distance increases. To address this, the RS-LiDAR-32 utilized advanced analysis of multiple return values, allowing it to output either the strongest, latest, or both signals depending on the situation. For this study, the dual echo mode was employed, which outputs both the strongest and final echo information simultaneously, resulting in a maximum output of 230,400 points per second.

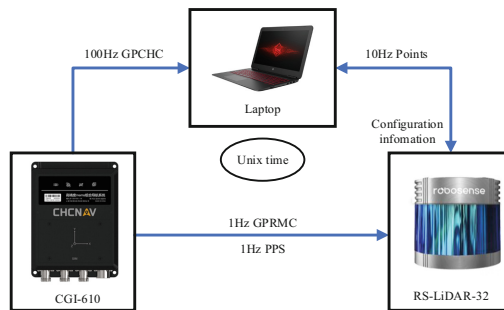
In this study, we define LiDAR data belonging to the same motor rotation from  $0^\circ$  to  $360^\circ$  as one LiDAR "frame". However, as the point clouds in a frame were collected at different times, motion distortion was generated due to the UGV motion during LiDAR data collection. When the UGV was moving straight at a speed of 0.5 m/s and the LiDAR was rotating at a rate of 600rpm, there was a difference of 0.05 m between the actual position of the earliest collected point cloud and the latest collected point cloud, as shown in Fig. 2(a). It was important to note that the front (+YL) environment of the UGV was a critical sensing range for the LiDAR. In addition, due to the fact that UGVs work in parks and had relatively low movement speeds, phase locking was used to reduce the distortion of laser point cloud motion. The LiDAR scans in an anticlockwise direction by default, with a starting scanning angle of  $0^\circ$ . When the Pulse-Per-Second (PPS) was triggered, the internal sensor rotated to the  $180^\circ$  position to initiate a new frame of scanning. This reduced the time interval of the point cloud in front of the UGV, as depicted in Fig. 2(b). By reducing the maximum time difference of the point cloud in front of the UGV from 100 ms to 50 ms, the motion distortion could be reduced from 0.05 m to 0.025 m, which was proximity to the LiDAR ranging accuracy and could be believed that the displacement distortion of the laser point cloud could be ignored. As the UGV's steering angle was small during normal driving conditions, rotation-induced point cloud distortion was disregarded in this study.



**Fig. 2.** Laser Rotation Direction: (a) No Phase Lock, Initial Angle was  $0^\circ$ ; (b) Phase Lock, Initial Angle was  $180^\circ$ .

### 2.3 Hardware Communication

The CGI-610 transmitted 100 Hz posed data to a laptop computer at a baud rate of 230400 through an RS232 to USB serial port cable. The data was defined by the GPCHC protocol, which includes vital information such as Greenwich time, UGV attitude angle, position, and movement speed. This reliable transmission system ensures accurate and efficient data transfer for optimal performance. In addition, GPRMC compliant messages and PPS simultaneously reached interface BOX through another serial port at a frequency of 1 Hz, achieving phase locking of internal motors and time synchronization of heterogeneous data, which was crucial for global point cloud mapping. The RS-LiDAR-32 was connected to the laptop through a CAT6 network cable, and point cloud data was transmitted through User Datagram Protocol (UDP). The laptop was installed with the Ubuntu system and ROS components, and the point cloud and field segmented pose data were saved in bag format by the ROS master according to Unix time in frames. Figure 3 summarized the connection and communication between among all components.

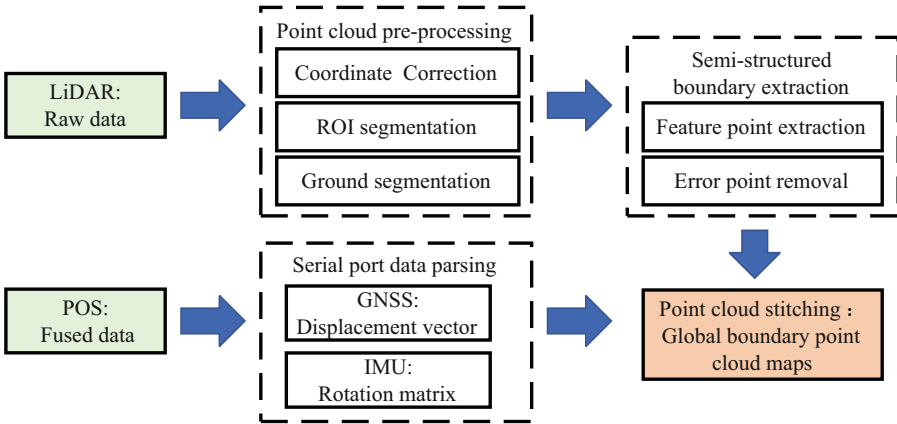


**Fig. 3.** Connections and Communications among all Components of the UGV-LiDAR System.

### 3 Methodology

#### 3.1 Overview

The problem of detecting semi-structured driving boundaries, such as low curbs in the park, could be broken down into two steps: point cloud pre-processing and semi-structured boundary extraction. Although low curbs in parks were usually planted uniformly, their intricate foliage made them harder to detect using the neighbouring point geometry feature method than the road curbs. To identify the driving boundary feature points, clusters of points are detected on both sides of the UGV’s forward direction based on the projection of the non-ground point cloud on the ground. Then, the protocol output from POS serial port is divided, the GNSS position and IMU attitude are analysed, and each frame of boundary point cloud is spliced to form a global point cloud map with the first frame of point cloud position as the origin. Figure 4 provided an overview of the proposed algorithm, that would be explain in detail below.



**Fig. 4.** Flowchart of the Proposed Semi-Structured Driving Boundary Detection and Mapping Algorithm.

#### 3.2 Algorithm Breakdown

##### 3.2.1 Coordinate Correction

The LiDAR data package only included parameters for horizontal rotation angle and distance. By combining these with the vertical angle of the laser transceiver, the point cloud could be converted from the polar coordinate system to the Cartesian coordinate system  $Ox_L Y_L Z_L$  to better display the 3D point cloud, as shown in Fig. 5. The relationship between the coordinate systems was illustrated in Eq. 1.

$$\begin{cases} X_L = \rho \cos(\omega) \sin(\alpha) \\ Y_L = \rho \cos(\omega) \cos(\alpha) \\ Z_L = \rho \sin(\omega) \end{cases} \quad (1)$$

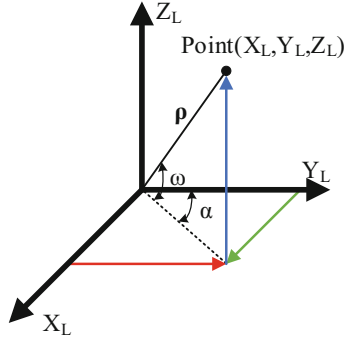


Fig. 5. LiDAR Polar and Cartesian Coordinate Mapping

where  $\rho$  was the measured distance,  $\omega$  was the vertical angle of the laser,  $\alpha$  was the horizontal rotation angle of the laser, and  $X_L$ ,  $Y_L$ ,  $Z_L$  were the coordinate value mapped to the Cartesian coordinate system.

The LiDAR was typically installed with a downward tilt of  $15^\circ$  to increase the point cloud density in the UGV's near environment. However, this tilt could impact the extraction of boundary feature points since it caused the  $X_L O Y_L$  plane to no longer be parallel to the ground. To address this issue, Eq. 2 was utilized to correct and transform the original point cloud.

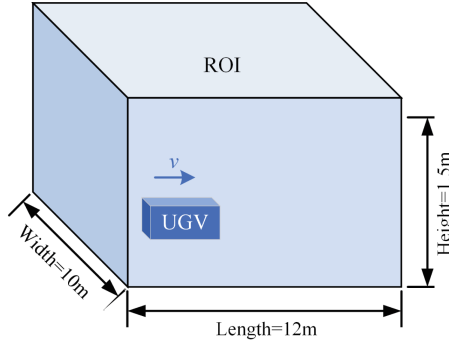
$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_0 & \sin \theta_0 \\ 0 & -\sin \theta_0 & \cos \theta_0 \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \quad (2)$$

where  $X_p$ ,  $Y_p$ , and  $Z_p$  were the coordinates after transformation, the  $XOY$  plane was parallel to the ground,  $\theta_0$  was the tilt angle of LiDAR installation.

### 3.2.2 Region of Interest Segmentation

Although the scanning range of 3D LiDAR was relatively large, unmanned ground radar only focuses on the feasibility of the driving area ahead during operation. To optimize processing time and increase precision, it was advisable to choose the area of interest for the corrected point cloud box, and eliminate any point cloud data outside of that region to complete the initial down-sampling process. Considering the travel speed and overall size of UGVs, a 3D cube method was used to select point cloud regions. In this study, the area of interest was determined to be centered around the geometric center of the LiDAR, with a total height of less than 1.5 m, including the upper and lower parts. The area extended 10 m forward, 2 m backward, and 5 m on each side. The schematic diagram of region of interest division was shown in Fig. 6.





**Fig. 6.** Region of Interest Segment Diagram

### 3.2.3 Ground Segmentation

Ground point clouds occupied a large proportion of the ROI area, so it was necessary to segment them for further down-sampling. The driving environment of UGV included weeds and gravel on the ground, and the attitude of LiDAR may change due to vibration and road conditions. Using a general direct filtering technique that solely relied on height information might not be sufficient to effectively remove all grounding points from the point cloud data. The use of Random Sampling Consistency (RANSAC) algorithm for ground point cloud segmentation had good applicability. The RANSAC algorithm was a robust algorithm used to estimate data model parameters from a dataset containing internal and external points by setting error thresholds and iterations. Assuming that the given dataset contained both inner and outer points, a subset of the original data was randomly selected at the beginning and assumed to be the model. Then, all other data were tested against the fitted model. When the data reached the error threshold, it was considered part of the consistent set of the model. Then, assuming that the randomly selected points were the model to be fitted, a consistent set was obtained, And the model with the highest number of points in the consistent set was judged as the result of fitting the model after a certain number of iterations.

In the algorithm design, first set the number of iterations of the algorithm as  $k$ , the distance error threshold as  $l$ , the total number of points of the point cloud as  $n$ . At the beginning, three points were randomly selected to form the ground to be fitted, and set the 3D coordinates of the three points as  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ ,  $(x_3, y_3, z_3)$ , then the fitted flat surface model was:

$$Ax + By + Cz + D = 0 \quad (3)$$

where the planar model parameters  $A$ ,  $B$ ,  $C$ , and  $D$  can be solved according to Eq. 4.

$$\begin{cases} A = (y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1) \\ B = (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1) \\ C = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1) \\ D = -(Ax_1 + By_1 + Cz_1) \end{cases} \quad (4)$$

Then the distance  $L$  from any point in space  $(x_0, y_0, z_0)$  to this plane was

$$L = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (5)$$

When the distance  $L$  between a point and the assumed plane satisfied  $L \leq l$ , the point was considered an internal point of the model. To determine the number of internal points,  $n-3$  points other than the initial 3 points were iterated through  $k$  iterations using a RANSAC method. The probability of obtaining accurate results increased with the number of iterations. The final ground model with the highest number of internal points was considered the best fit.

### 3.2.4 Feature Point Extraction

After down-sampling including ROI segmentation and ground segmentation, non-ground point cloud data such as shrubs were left behind, forming driving boundaries that affect the passage of UGVs. However, due to the complex structure of shrub leaves, it can be challenging to accurately extract the boundaries from the scanned point clouds, which lack clear geometric features. To address this issue, the point cloud is first projected and dimensionally reduced, and then feature points that can effectively represent the driving boundary are extracted.

All non-ground points were projected onto the plane represented by Eq. 3 when projecting point clouds. The resulting projected point cloud contour were irregular. To simplify this contour, we divided the area in front of the UGV into intervals of  $\Delta t$ . Then, we selected the closest points on both sides of the UGV. This was illustrated in Fig. 7. The points with the largest  $X_P$  in the left area of  $Y_P$  and the points with the smallest  $X_P$  in the right area of  $Y_P$  were extracted.

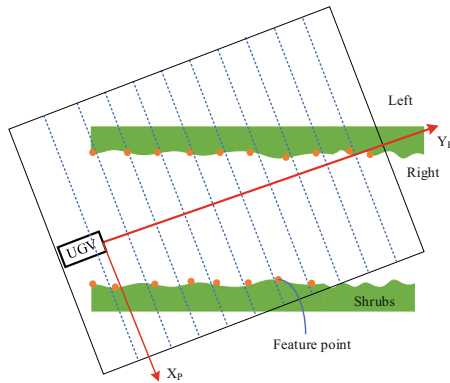
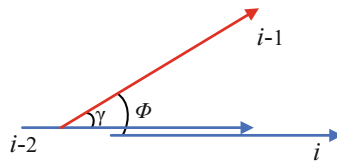


Fig. 7. Schematic Diagram of Boundary Feature Point Extraction

### 3.2.5 Error Point Removal

Due to the gaps between the leaves, the laser beam might pass through the bushes and shine on the inside or on the other side, which could result in significant differences

in the morphology of local areas compared to other positions due to the growth of the bushes. To address this issue and eliminate any detected error boundary feature points, this study employed a two-step method to remove outliers. In the first step, the RANSAC algorithm was used to iteratively fit the optimal line model of single frame feature points. The iterative process involves randomly selecting two points from the set of data points, and calculating the parameters of the line passing through these points. The distance between each point and the line was then calculated, and those points whose distance exceeded the set threshold were considered outliers and removed from the data set. The line was then re-fitted to the remaining inliers, and the process was repeated for the specified number of iterations or until convergence was achieved. This involved retaining points that matched the line model while removing any points that fall outside of the model. The second step involved conducting inter-frame evaluation by using the linear model of each point cloud frame. The cosine theorem was applied to determine the angle between two frames based on their linear model parameters. When the model parameters of the  $i$  frame were fitted, the angle between the straight line of the  $(i-1)$  frame and the  $(i-2)$  and  $i$  frames respectively was found according to the cosine theorem, and if it was greater than the threshold value, it means that the possibility of wrong extraction of feature points in that frame was higher and is eliminated, as illustrated in Fig. 8.



**Fig. 8.** Boundary Line Angle Method for Removing Erroneous Feature Points

### 3.2.6 Point Cloud Splicing and Mapping

The boundary feature points extracted from the laser point cloud were in a coordinate system with their own geometric center as the origin. To ensure the safe driving of UGVs in the park, a global map was constructed to improve the density of the point cloud. There were two different approaches to creating a global map using POS data integration. The first method involved transforming all point clouds into a map with the Earth's center of mass as the origin. The second method involved using the position of the first frame point cloud as the coordinate origin and then concatenating each subsequent frame point cloud to that position. The latter had small coordinate values and high accuracy, and this study adopted this method. The POS was comprised of two components: GNSS and IMU. The GNSS system calculated the position of the UGV in the Earth coordinate system, while the IMU system calculated the attitude angles (Yaw, Roll, Pitch) of the UGV in the local horizontal coordinate system. The point cloud frames were translated using the corresponding GNSS positions and UTM projection to determine the difference, followed by obtaining the rotation matrix for transforming the point cloud to the geographic coordinate system using the attitude angle output provided

by the IMU. The  $X_{East}$ ,  $Y_{North}$ , and  $Z_{Altitude}$  directions were defined as the east, north, and elevation directions, respectively.

Due to the fact that the latitude, longitude, and altitude data output by POS are coordinates under the Earth's ellipsoid, representing the position of the IMU, rather than the center of the LiDAR scanner. Assuming the coordinates of latitude, longitude, and height transformed by UTM projection were  $(X_I, Y_I, Z_I)$ , the formula for converting laser point cloud coordinates to global map coordinates was:

$$\begin{bmatrix} X_{East} \\ Y_{North} \\ Z_{Altitude} \end{bmatrix} = R \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \quad (6)$$

where  $R$  was a rotation matrix derived from IMU attitude angles (Yaw, Roll, Pitch) in Euler angle format converted from quaternions,  $(d_x, d_y, d_z)$  represented the position of the geometric center of the LiDAR relative to the center of the IMU, and  $(\Delta x, \Delta y, \Delta z)$  represented the offset of the IMU center from the global map origin.

## 4 Semi-structured Boundary Detection Tests

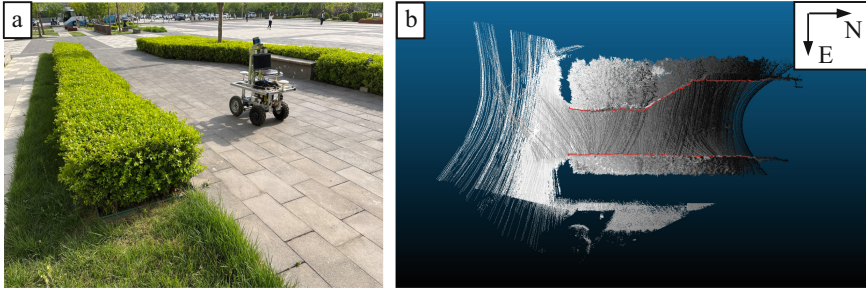
### 4.1 Data Collection

We collected data in a park in Beijing's Haidian District in April 2023 to assess the effectiveness and accuracy of the UGV-LiDAR system for semi-structured boundary extraction algorithm. The LiDAR and IMU position error of our UGV-LiDAR system were calibrated using automated algorithms in defined scenarios, while the lever arm offset between the GNSS receiver and LiDAR was manually measured for accuracy. For data collection, we carefully selected two scenarios. In the first scenario, trimmed shrubs were planted on both sides of the UGV, with small and dense leaves, and the distance between them gradually decreased. In the second scenario, only untrimmed shrubs were planted on the left side of the UGV, with leaves growing in strips and varying heights. In both cases, the UGV travelled at a speed of 0.5–0.6 m/s on the curve within the region. The data was saved as a separate bag file. In addition, the basic geometric dimensions of the shrubs were manually measured as true values for comparison with the automatically extracted boundaries to evaluate the algorithm performance.

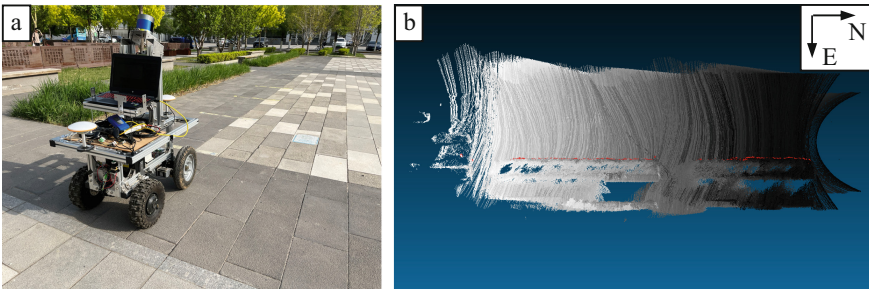
### 4.2 LiDAR Data Processing

The LIDAR driver package was used to complete the mapping of the point cloud coordinate system, and the subsequent steps were implemented in the ROS system using the C++ programming language. These steps included point cloud segmentation, boundary extraction, and data fusion. During the ground segmentation stage, the RANSAC algorithm was set to run for 100 iterations with a distance threshold of 0.1 m. Feature points were extracted at distance intervals of 0.2 m. For fitting boundary feature points with RANSAC straight lines, 20 drops were set with a distance threshold of 0.1 m. Any boundary line angle extracted between frames greater than  $10^\circ$  was considered an incorrect feature point.

The transformed point cloud was converted into PCD format, which was a format defined by the PCL library for point clouds. The visualization was done using CloudCompare 2.11. Additionally, the algorithm's processing time for each frame of point cloud was recorded during data collection. Figure 9 depicted the data collection scene and the resulting point cloud generated by the UGV-LiDAR system, which captured shrubs of varying widths on both sides. In contrast, Fig. 10 portrayed the data collection scene and point cloud generated by the system, which captured dispersed shrubs on one side.



**Fig. 9.** Scenario 1: The Distance Between the Shrubs on both Sides Gradually Becomes Closer. (a) UGV-LiDAR Data Collection Site. (b) Global Point Cloud Map within ROI (Grey) and Boundary Point Cloud (Red).



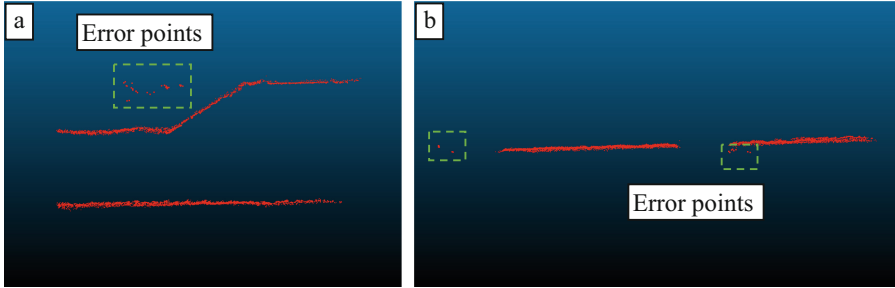
**Fig. 10.** Scenario 2: Left Shrub Segmented Continuous Planting. (a) UGV-LiDAR Data Collection Site. (b) Global Point Cloud Map within ROI (Grey) and Boundary Point Cloud (Red).

### 4.3 Experimental Results

#### 4.3.1 Boundary Detection Accuracy

In this analysis, 100 processed point clouds were selected from two different scenarios to construct point cloud maps to evaluate the accuracy of the boundary detection algorithm, as shown in Fig. 11. In Scenario 1, the algorithm accurately detected 3857 left boundary points, and the error detection point is 0, with an accuracy rate of 100%. On the right

boundary, 3150 points were detected, with 33 errors and an accuracy rate of 98.95%. In scene 2, a total of 7764 boundary point was detected on the left side of the UGV, and only 23 error detection points were detected, with an accuracy of 99.70%. The detection error might be caused by sensor noise or laser beam scanning through bushes to a distance.



**Fig. 11.** Boundary Point Cloud. (A) Scenario 1; (B) Scenario 2.

Using the point cloud map constructed in Scenario 1, the width of shrubs on both sides was measured based on the position of manual measurements, and the results were recorded in Table 2. The average error in the width of the ten measured positions was found to be 0.033 m, which accurately reflects the width of the area restricted by the driving boundary in the actual environment. By collecting these measurements, a more accurate depiction of the physical environment can be obtained, which can then be used to develop more effective driving strategies for the UGV. This information can aid in the creation of more efficient and reliable autonomous systems for various applications.

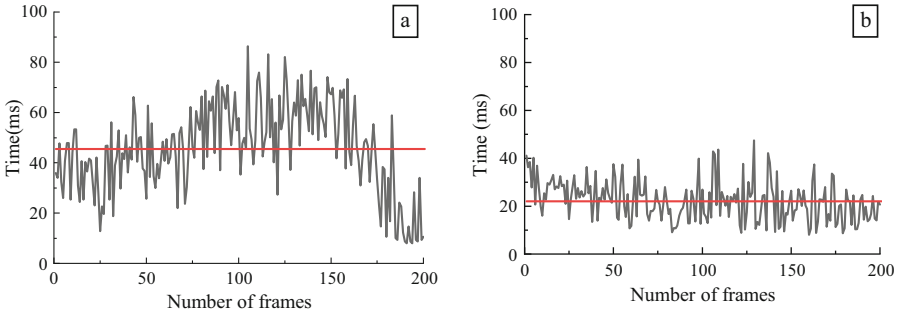
**Table 2.** Comparison of Boundary Widths on both Sides of the Map in Scenario 1 with Actual Measurements

Number	1	2	3	4	5	6	7	8	9	10	Mean
Truth/m	2.983	2.863	2.792	2.871	2.788	2.998	3.482	4.154	4.770	4.719	–
Map/m	2.887	2.771	2.781	2.734	2.715	2.970	3.594	4.249	4.703	4.675	–
Error/m	0.096	0.092	0.011	0.137	0.073	0.028	−0.12	−0.095	0.067	0.044	0.033

### 4.3.2 Runtime of Proposed Method

The boundary detection algorithm for the driving area is mainly used for the application of UGV during the driving process. Therefore, it is important for algorithms to run effectively in real-time. The running time of the algorithm proposed in the second section was calculated, as shown in Fig. 12. In scenario 1, the average frame time was 45.7 ms, in scenario 2, the average frame time was 22.4 ms, and the 3D LiDAR scanning frequency was 10 Hz; Therefore, the algorithm can run completely in real-time. Although the

operation of the boundary detection process mainly involves 3D point clouds, predictive search methods were used by us to narrow the search range and greatly reduce the calculation time per frame. It is obvious that when there are boundaries on both sides of a UGV that affect traffic, the calculation time is twice as long as when there are boundaries on one side, because the number of point clouds calculated is twice.



**Fig. 12.** Runtime of the Proposed Semi-structured Boundary Detection Algorithm. (a) Scenario 1; (b) Scenario 2.

## 5 Conclusion and Future Work

In this work, we proposed a specialized algorithm to quickly extract semi-structured driving area boundaries from the 3D LiDAR data of UGVs working in the park, and integrate GNSS/IMU data to construct a global point cloud map of the driving boundaries. The boundary extraction performance of the algorithm is tested in two scenarios. In the scene with semi-structured shrubs on both sides, the accuracy of boundary point detection is 98.95%, the average distance error of green belts on both sides is 0.033m, the average processing time is 45.7 ms/frame, and the average processing time is about 22.4 ms/frame. In the scene of planting low shrubs on one side, the accuracy of boundary point detection is 99.70%, and the average processing time is about 22.4 ms/frame. Overall, the boundary detection algorithm shows high accuracy in both cases, with only a small number of error detection points, and meet the real-time requirements.

Currently, we are researching a point cloud registration method for unstructured driving boundaries to cope with the environment where the signal of the integrated navigation system is obstructed by tall buildings or trees, and to achieve boundary extraction without the assistance of GNSS/IMU pose information, and thus achieve path planning for UGVs.

## References

1. Luettel, T., Himmelsbach, M., Wuensche, H.-J.: Autonomous ground vehicles—concepts and a path to the future. Proc. IEEE **100**(Special Centennial Issue), 1831–1839 (2012)

2. Raj, R., Kos, A.: A comprehensive study of mobile robot: history, developments, applications, and future research perspectives. *Appl. Sci.* **12**(14), 6951 (2022)
3. Xu, F., Wang, H., Hu, B., et al.: Road boundaries detection based on modified occupancy grid map using millimeter-wave radar. *Mob. Netw. Appl.* **25**, 1496–1503 (2020)
4. Nan, Z., Wei, P., Xu, L., et al.: Efficient lane boundary detection with spatial-temporal knowledge filtering. *Sensors* **16**(8), 1276 (2016)
5. Han, J., Kim, D., Lee, M., et al.: Enhanced road boundary and obstacle detection using a downward-looking LiDAR sensor. *IEEE Trans. Veh. Technol.* **61**(3), 971–985 (2012)
6. Hata, A.Y., Ramos, F.T., Wolf, D.E.: Monte Carlo localization on Gaussian process occupancy maps for urban environments. *IEEE Trans. Intell. Transp. Syst.* **19**(9), 2893–2902 (2017)
7. Panev, S., Vicente, F., De la Torre, F., et al.: Road curb detection and localization with monocular forward-view vehicle camera. *IEEE Trans. Intell. Transp. Syst.* **20**(9), 3568–3584 (2018)
8. Cheng, M., Zhang, Y., Su, Y., et al.: Curb detection for road and sidewalk detection. *IEEE Trans. Veh. Technol.* **67**(11), 10330–10342 (2018)
9. Fernandez, C., Fernandez-Llorca, D., Sotelo, M.A.: A hybrid vision-map method for urban road detection. *J. Adv. Transp.* **2017** (2017)
10. Demir, S.Ö., Ertop, T.E., Koku, A.B., et al.: An adaptive approach for road boundary detection using 2D LiDAR sensor. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 206–211. IEEE (2017)
11. Wijesoma, W.S., Kodagoda, K.R.S., Balasuriya, A.P.: Road-boundary detection and tracking using lidar sensing. *IEEE Trans. Robot. Autom.* **20**(3), 456–464 (2004)
12. Romero, L.M., Guerrero, J.A., Romero, G.: Road curb detection: a historical survey. *Sensors* **21**(21), 6952 (2021)
13. Sun, P., Zhao, X., Xu, Z., et al.: A 3D LiDAR data-based dedicated road boundary detection algorithm for autonomous vehicles. *IEEE Access* **7**, 29623–29638 (2019)
14. Huang, R., Chen, J., Liu, J., et al.: A practical point cloud based road curb detection method for autonomous vehicle. *Information* **8**(3), 93 (2017)
15. Rodriguez-Cuenca, B., Garcia-Cortes, S., Ordóñez, C., et al.: An approach to detect and delineate street curbs from MLS 3D point cloud data. *Autom. Constr.* **51**, 103–112 (2015)