IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2024, accepted 24 July 2024, date of publication 2 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3437784

RESEARCH ARTICLE

A Novel Obstacle Detection Method in Underground Mines Based on 3D LiDAR

PINGAN PENG, JIN PAN, ZIYU ZHAO, MENGNAN XI, AND LINXINGZI CHEN
School of Resources and Safety Engineering, Central South University, Changsha 410083, China

Corresponding author: Jin Pan (225512110@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52104170, in part by the Science and Technology Innovation Program of Hunan Province under Grant 2023RC3069, in part by the National Key Research and Development Program of China under Grant 2022YFC2904105, and in part by the Fundamental Research Funds for the Central Universities of Central South University under Grant 2023ZZTS0134.

ABSTRACT In mine operations, the safe operation of transportation equipment is crucial to ensure the safety of miners and the efficiency of mine production. However, it is notable that there is little research on perception technology for unstructured environments such as underground mining tunnels. The underground mining environment is characterized by its intricate nature, with narrow passageways, dim lighting, and complex spatial topological structures. Large-scale mining trucks operating in such environments have a restricted field of view and pose a serious safety hazard. In this paper, we propose an underground mining obstacle detection method based on 3D light detection and ranging (LiDAR) technology to augment the environmental perception capabilities of mining vehicles. This method uses point cloud data collected by LiDAR as input, employing an improved random sample consensus (RANSAC) to segment rugged ground points. Additionally, an innovative point cloud processing module for tunnel walls and the application of Euclidean clustering and obstacle recognition strategies ensure accurate obstacle detection. Experimental results demonstrate that the proposed method achieves a detection accuracy of over 95% within a 50-meter region of interest, and the running time is kept within 0.14 seconds on an ordinary computer. The effectiveness of the proposed method is discussed across varying distances, numbers, and tunnel types, revealing satisfactory outcomes and robust applicability. The proposed efficient method meets the requirements of underground mining truck obstacle detection, making a substantial contribution to underground unmanned production.

INDEX TERMS Obstacle detection, point cloud, underground mine, LiDAR.

I. INTRODUCTION

Safety, efficiency, and environmental concerns have come to the forefront of underground mining operations [1], [2], [3]. In mine operations, the safe operation of transportation equipment is crucial to ensure the safety of miners and the efficiency of mine production. However, the underground environment is complex and variable, with numerous potential obstacles such as falling rocks and walking workers. When large mining trucks operate in narrow underground mines, their front view is limited, posing significant risks during transportation. Additionally, as mining depth increases and mining scales expand, underground transportation routes have become more complicated. Traditional manual inspection and monitoring methods are inefficient and struggle to detect potential obstacles and hazards promptly. With the gradual adoption of automation and intelligent technology in mining operations, ensuring the safe operation of transportation equipment in unmanned conditions has become an urgent problem [4], [5], [6], [7]. Therefore, applying object detection technology to underground mining transportation is a vital task [8]. This technology can achieve

The associate editor coordinating the review of this manuscript and approving it for publication was Guolong Cui.

timely detection and early warning of obstacles, prevent accidents, and enhance mine safety. It can also improve the automation of transportation equipment, reduce the cost of manual inspection, and boost overall production efficiency and economic benefits.

Currently, there is a lack of mature technical means for obstacle detection in underground mines. Road obstacle detection has been developed more maturely in recent years [9], [10], but underground mines are different from structured roads, which have rugged road surfaces, dark environments, and complex spatial topologies. Structured road detection methods directly applied to the mining environment will have problems such as incompatible algorithms, excessive or insufficient segmentation, and poor recognition results. Therefore, it is necessary to propose an obstacle detection method that performs well in underground mines [11].

The sensing devices used for obstacle detection are mainly cameras and light detection and ranging (LiDAR). Cameras are easily affected by changes in ambient light [5], [12], [13], [14], [15], while LiDAR has outstanding performance in acquiring object features, achieving detection accuracy, and performing well in low-light environments [16], [17], [18], [19], [20]. Considering the dark environment of an underground tunnel, this paper focuses on the obstacle detection method based on LiDAR.

The two main categories of LiDAR-based obstacle detection methods are traditional and deep learning based methods. In the traditional methods, Li et al. introduced an obstacle detection method based on multi-frame point cloud fusion and ground plane estimation [21]. Asvadi et al. presented a 3D perception system incorporating two key modules: ground surface estimation using a segmented plane fitting algorithm and random sample consensus (RANSAC) method, and a model for detecting static and moving obstacles employing discriminative analysis and ego-motion information [22]. Tian et al. introduced a k-NN algorithm for obstacle detection and labeling [23]. Le and Trân proposed a point cloud segmentation method synthesizing RANSAC and adaptive Euclidean clustering algorithms, and its effectiveness was verified in real data [24]. Most of these methods are mainly applied to flat traffic roads and campus trails, and the results in unstructured environments such as underground mines are hardly satisfactory [25].

In recent years, scholars have also commenced research on obstacle detection methods based on deep learning. Ben-Shabat et al. proposed the utilization of cumulative convolutional neural networks for a 3D point cloud, employing a grid for structural representation [26]. Additionally, a novel 3D point cloud representation, the 3D modified Fisher vector, was introduced. Beltrán et al. presented BirdNet, an approach for obstacle detection and classification that involves projecting point cloud data into a cell code for bird's-eye projection and leveraging a convolutional neural network to estimate obstacle position and direction [27]. Li extended the detection technique based on fully convolutional networks to the 3D domain, applying it to point cloud data [28]. While this method exhibits significant performance improvements over previous approaches, it is noteworthy that the detection speed is comparatively slow. Nonetheless, these approaches require substantial training datasets, and their detection performance is reliant on the comprehensiveness of the training data. Furthermore, they impose significant computational demands, impacting the computational resources required for the operation of other autonomous driving systems.

The efficient LiDAR-based obstacle detection method proposed in this paper integrates the advantages of previously mentioned methods while addressing their limitations to eliminate over-segmentation and under-segmentation issues. This study uses point cloud data collected by LiDAR as input, employing an improved RANSAC to segment rugged ground points. Additionally, an innovative point cloud processing module for tunnel walls and the application of Euclidean clustering and obstacle recognition strategies ensure accurate obstacle detection. The main contributions of this paper are as follows:

1. Considering the characteristics of the underground mine tunnel environment, an efficient LiDAR-based obstacle detection method for underground mines is proposed, with its real-time performance and feasibility confirmed through experiments.

2. The innovative inclusion of a point cloud processing module for tunnel wall surfaces enables self-adaptation to different types of tunnels.

The rest of the paper is organized as follows. Section II systematically describes the processing flow of the proposed method. Section III details the principles of the algorithms involved. Section IV verifies the method's effectiveness through experiments. Section V discusses the proposed method, and Section VI summarizes the entire paper.

II. SYSTEM OVERVIEW

The entire processing flow of the efficient obstacle detection method is illustrated in Fig. 1. It comprises three modules: point cloud preprocessing, point cloud segmentation, and obstacle detection with bounding box labeling. In the first module, we preprocess the input point cloud by extracting the region of interest and performing voxel grid downsampling to significantly reduce processing time caused by the large volume of data. The second module presents a method for segmenting the point cloud of both the ground and the tunnel wall surfaces. For the ground point cloud segmentation, an optimized RANSAC method is employed to address the issue of discontinuous point clouds that cannot be fitted due to the rugged roadway. For the wall point cloud segmentation, a grid map-based method is used to extract and fit the tunnel boundary points, thus achieving the segmentation of the tunnel wall points. In the third module, Euclidean clustering is utilized along with the tunnel environment obstacle recognition constraint method to achieve obstacle detection, outputting the 3D information of the obstacle in the form of a bounding box.

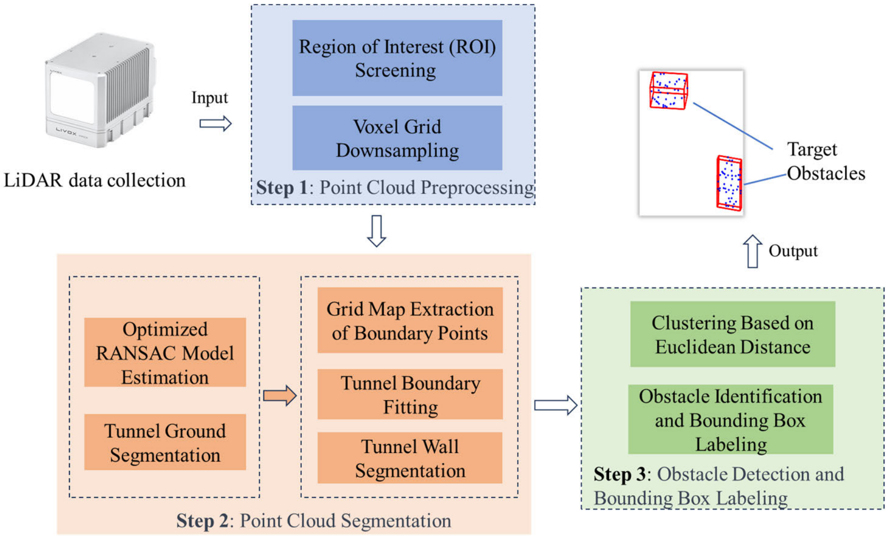

Input
LiDAR data collection
Region of Interest (ROI) Screening
Voxel Grid Downsampling
Step 1: Point Cloud Preprocessing
Target Obstacles
Output
Optimized RANSAC Model Estimation
Tunnel Ground Segmentation
Grid Map Extraction of Boundary Points
Tunnel Boundary Fitting
Tunnel Wall Segmentation
Step 2: Point Cloud Segmentation
Clustering Based on Euclidean Distance
Obstacle Identification and Bounding Box Labeling
Step 3: Obstacle Detection and Bounding Box Labeling

FIGURE 1. **Algorithm framework.**

III. METHOD

A. POINT CLOUD PREPROCESSING

Due to the extensive scanning range of the 3D LiDAR and the fact that mine trucks only need to perceive the forward driving area, the LiDAR-collected point cloud is initially partitioned into a region of interest. Points within this region are then subjected to subsequent processing to reduce processing time and improve detection accuracy. The region of interest chosen for this study is a 3D cube in space, as shown in Fig. 2.

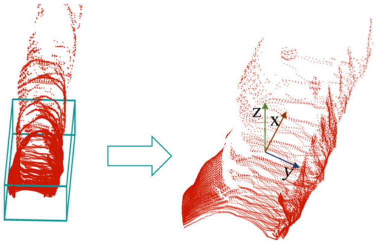


FIGURE 2. **The region of interest.**

Upon selecting the region of interest, the number of points in the point cloud remains substantial, necessitating a sampling process. Two prevalent techniques for downsampling include voxel downsampling and random downsampling. The latter is marked by its simplicity and superior processing speed. However, voxel downsampling operates within a spatial grid, thereby significantly preserving the spatial intricacies of the data [29], [30]. Given the inherent geometric attributes of the point cloud, voxel downsampling undoubtedly emerges as the more fitting choice, reducing data volume while concurrently preserving the salient features of the point cloud.

The fundamental concept behind voxel downsampling involves partitioning the three-dimensional space of a point cloud into closely adjacent spatial grids, known as voxels. Each voxel encapsulates all the points within its confines, approximating their presence through specific points of significance within the voxel, such as its centroid and barycenter. The illustrative diagram of voxel segmentation is depicted in Fig. 3.

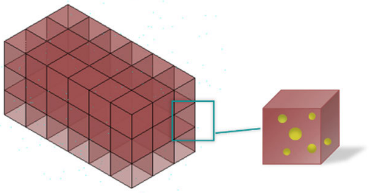


FIGURE 3. **Diagram of the voxel grid segmentation.**

In this paper, the center of gravity of each voxel is selected to represent all points in the grid. If there are no points in the voxel, the data closest to the center of gravity in the small grid is used instead. After the processing of voxel downsampling, the number of point cloud data will be significantly reduced and the spatial characteristics of the point cloud will be well preserved, effectively accelerating the subsequent processing.

B. AN OPTIMIZED RANSAC SEGMENTATION OF GROUND POINT CLOUD

The spatial characteristics of the ground point cloud are related to the environment of the underground tunnel. The tunnel with a long service life is in better repair and the ground is flat, while the ground in most tunnels with a short service life is generally irregular. Therefore, the data point cloud scanned by LiDAR also shows different characteristics. In the former case, the ground point cloud is almost on the same plane and continuous, while the irregular ground point cloud shows some intermittent occlusion and so on. In addition, the point cloud in space shows dense near point cloud and sparse far point cloud, which are also the characteristics of LiDAR scanning.

Currently, several common methodologies exist for segmenting ground point cloud, including height-based, plane-fitting, gradient-based, and random sampling-based point cloud segmentation approaches [31]. Employing a sole height constraint for ground segmentation in uneven roadways often leads to issues of either excessive segmentation or insufficient segmentation. Plane fitting necessitates prior acquisition of a rough estimate of the ground point cloud, whereas the RANSAC method offers superior flexibility and fitting outcomes.

However, due to the aforementioned characteristics of the tunnel, the simple application of RANSAC for ground point cloud segmentation is not as effective as desired. There may arise situations where a large-area ground point cloud cannot be accurately segmented, consequently affecting the subsequent obstacle detection module. Therefore, a finer segmentation is required to achieve better processing results. Taking these factors into account, an optimized RANSAC method is proposed for underground mine ground segmentation.

In irregular ground scenarios, the point clouds obtained from laser scanning can exhibit discontinuities. Consequently, an optimized version of the RANSAC method is utilized here. This method divides the point cloud into multiple segments by assessing whether the angular difference of the normal vectors of the points reaches a set threshold.

Subsequently, planar fitting is conducted on each segment separately. By doing so, it addresses the issue of inconsistent point clouds that cannot be effectively fitted due to roadway irregularities, thereby improving the accuracy and robustness of the plane fitting process. The before-and-after comparison of the method optimization is illustrated in Fig. 4.

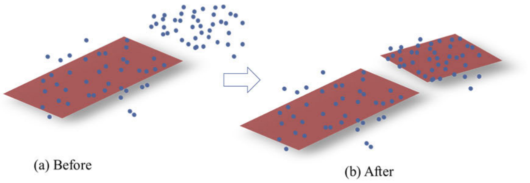


FIGURE 4. Comparison before and after method optimization.

The following are the steps of the ground point cloud removal algorithm.

1) Calculate the normal vector at each point using the function in the PointCloudLibrary library.

2) Start from the seed point and add it to the current segmentation region. Iterate through the neighboring points of the current region and for each neighboring point p_j, compute the angular difference θ_{ij}, between its normal vector n_j, and the seed point's normal vector n_i.

3)If θ_{ij} is less than the threshold $\theta_{threshold}$, add the point p_j to the current region and mark it as visited. Repeat the above steps until no new point can be added to the current region or the preset stop condition is reached.

4) Within each segment, apply the RANSAC method for ground surface fitting. Specify the iteration count as T, a distance threshold as a and compute the point-to-plane distance d Points with distances less than the threshold a are considered inliers. Among all iterations, retain the model with the highest count of inliers within the defined iteration range. The formula for calculating distance d is as follows:

$$d = \frac{|ax + by - z + c|}{\sqrt{a^2 + b^2 + 1}} \tag{1}$$

where (x, y, z) represents the data coordinate and (a, b, c) represents the parameters of the fitted plane.

5) Independently record both outlier and inlier data, subsequently generating the output dataset. The collection of outlier data represents non-ground point cloud elements, while conversely, the collection of inlier data pertains to ground point cloud elements.

C. TUNNEL WALL POINT CLOUD SEGMENTATION

The objective of this particular step is to eliminate the point cloud data corresponding to the walls of the mine tunnel, thereby advancing the acquisition of obstructive point cloud data and ultimately augmenting the precision of the detection process. Figure 8 illustrates a horizontal top-down view subsequent to the removal of ground-related point cloud data. From the depiction, it becomes apparent that the point cloud data representative of the tunnel walls exhibits a continuous and banded distribution along the horizontal plane. Leveraging this distinct characteristic, the present study employs a method for segregating the wall-associated point cloud data. This approach entails an initial process of curvilinear fitting to the external surface point cloud data, followed by an inward displacement to ascertain the inner surface curve. Subsequently, the wall-related point cloud data is expunged. The algorithm pseudo code is shown in Fig. 5.

1) ESTABLISHING A GRID MAP

This approach bears a resemblance to voxel grid processing, differing in that voxel grids are three-dimensional, whereas this grid is two-dimensional. Initially, the parameters of the grid map are established: the total length L and width W correspond to the length and width of the region of interest in the preprocessing stage. The side length of the small grid within the grid, denoted as $a \times a$, is determined. Consequently, the resolution of the grid map is defined as L/a pixels in length and W/a pixels in width. Subsequently, an iteration through all the point cloud data takes place, assigning each point to its respective grid position (n_x, n_y) as computed by the following formula:

$$n_x = [(x_i - x_{\min}) / a] \tag{2}$$

$$n_y = [(y_i - y_{\min}) / a] \tag{3}$$

Both n_x and n_y are rounded down.

```
Algorithm: Tunnel Wall Point Cloud Segmentation
Input: Point cloud $ptCloud$, grid parameters $L, W, a$, scaling factor $\omega$, offset distance $d$
Output: Cleaned point cloud $nowall\_ptcloud$
Initialize gridMap to an empty map
for each point $(x_i, y_i, z_i)$ in ptCloud do
    $n_x = (x_i - x_{\min}) / a$
    $n_y = (y_i - y_{\min}) / a$
    gridMap[nx, ny].append($(x_i, y_i, z_i)$)
end for
Initialize scaledPointCloud to an empty list
for each point $(x_i, y_i, z_i)$ in ptCloud do
    scaledPointCloud.append($(\omega x_i, y_i, z_i)$)
end for
clusters = performClustering(scaledPointCloud)
boundaryClusters = getTwoLargestClusters(clusters)
upperBoundaryCurve,lowerBoundaryCurve=fitBoundaryCurves(boundaryClusters)
Initialize cleanPointCloud to an empty list
$y_1$'= offsetCurve(upperBoundaryCurve, -$d$)
$y_2$'= offsetCurve(lowerBoundaryCurve,$d$)
for each point $(x_i, y_i, z_i)$ in ptCloud do
    if $(y_i - y_1'(x_i))(y_i - y_2'(x_i)) >= 0$
        nowall_ptcloud.append($(x_i, y_i, z_i)$)
    end if
end for
Return $nowall\_ptcloud$
```

FIGURE 5. Tunnel wall point cloud segmentation algorithm.

2) EXTRACTING THE BOUNDARY POINT CLOUD

Lidar can only scan the environment inside the tunnel, and the environment outside the tunnel walls will not be scanned. Taking advantage of this characteristic, after generating a point cloud grid map, we traverse from both ends of the grid map and obtain the first grid in each row that contains point clouds, which is defined as the boundary point cloud grid, as shown in Fig. 6. Finally, all grids containing boundary point clouds on both sides are obtained.

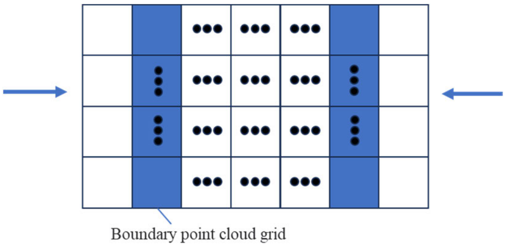


FIGURE 6. Boundary point cloud data extraction.

The point cloud obtained by this method is affected by factors such as obstacles and uneven surfaces, leading to issues like boundary discontinuities, false boundaries, and noise. Therefore, it is necessary to further refine the initial point cloud mesh. There is a clear distinction between boundary points and non-boundary points: the boundary point cloud extends along the x-axis but is discontinuous, while non-boundary points are shorter in the x-axis direction. Based on this feature, clustering technology is applied to the point cloud data to differentiate between boundary points and non-boundary points.

To eliminate the impact of boundary discontinuities on the results, the following preprocessing is performed on the x-axis coordinates of the point cloud before clustering:

$$p(x_i, y_i) \to p'(\omega x_i, y_i) \text{ where } 0 < \omega < 1. \tag{4}$$

Here, ω denotes a weighting factor. A smaller ω enhances the robustness of clustering against local x-axis discontinuities. This means that as ω decreases, the clustering becomes more stable in aggregating discontinuous boundary point clouds along the x-axis into a single category while distinguishing between y-axis boundary point clouds and non-boundary point clouds, resulting in greater stability.

Subsequently, based on the characteristic that the x-axis length of the boundary point clouds is significantly greater than that of the non-boundary points, we calculate the length L on the x-axis for different clusters, where $L = x_{\max} - x_{\min}$. This calculation allows us to identify the two clusters with the largest L values, corresponding to the left and right boundary point clouds. At this stage, the boundary point cloud extraction process is complete.

3) BOUNDARY FITTING

The obtained boundary point cloud from both sides is subjected to quadratic curve fitting, represented as $y = ax^2 + bx + c$, where a, b and c represent the coefficients of the quadratic, linear, and constant terms, respectively.

The overdetermined equation system $Ax = y$ satisfies the following:

$$x = \begin{bmatrix} x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}, A = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \tag{5}$$

When c is greater than zero, it signifies that the fitted boundary curve represents the upper boundary ($y > 0$), whereas when c is less than zero, it corresponds to the lower boundary ($y < 0$). To facilitate computer-based differentiation, we will label the upper and lower boundary fitted curves as y_1 and y_2 respectively.

4) SEGMENTATION OF TUNNEL WALL POINT CLOUD

The boundary curves y_1 and y_2 representing the outer boundary of the tunnel wall have been obtained in the previous step. The inner boundary curve of the tunnel can be obtained by equidistant offset of the outer boundary curve: the lower boundary moves upward and the upper boundary moves downward. The curves obtained by the offset are denoted as y_1' and y_2'.

Given a point (x_0, y_0, z_0), if this point falls between y_1' and y_2', it is considered to be a non-wall point cloud. Conversely, if it falls outside this range, it is categorized as a wall point cloud. This approach allows us to effectively distinguish between wall and non-wall point clouds in the tunnel environment.

Therefore, we establish the following constraint condition:

$$(y_i - y_1') \times (y_i - y_2') \geq 0 \tag{6}$$

If this condition is met, the point is considered a wall point cloud and is subsequently removed. Conversely, if the condition is not met, the point is retained as a non-wall point cloud. This approach allows us to effectively filter and remove wall point clouds from the dataset.

D. OBSTACLE DETECTION AND BOUNDING BOX LABELING

1) POINT CLOUD CLUSTERING

Obstacle point clouds typically appear in the form of clustered point clouds with closely spaced points. There is usually a considerable separation between different clusters representing distinct obstacles. Given this characteristic, we employ a clustering algorithm based on Euclidean distance to effectively identify and differentiate between various obstacle clusters within the point cloud data. The fundamental idea of this algorithm is to calculate the Euclidean distance between points to determine whether they belong to the same cluster and use labeling to identify these clusters. For two points, $p_i(x_i, y_i, z_i)$ and $p_j(x_j, y_j, z_j)$, the Euclidean distance calculation formula is as follows:

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{7}$$

The algorithm's details are as follows:

Step 1**:** Initially, validate and process the input point cloud to remove any invalid points.

Step 2**:** Use the Euclidean distance between points to identify neighboring points within the point cloud, enabling segmentation. Parallel search methods can be employed to enhance performance.

Step 3**:** Employ labels to identify the cluster to which each point belongs. Subsequent processing includes the removal of clusters that do not meet specified criteria.

Step 4**:** Return the segmented point cloud and the count of clusters.

The schematic diagram of the clustering principle is shown in Fig. 7.

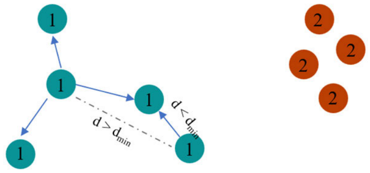


FIGURE 7. Schematic diagram of clustering principle.

2) OBSTACLE IDENTIFICATION AND BOUNDING BOX LABELING

Due to the complexity of the tunnel environment, the clustering results obtained in the previous section contain not only obstacle point clouds but also some noise and non-obstacle point cloud clusters. Therefore, further optimization of the results is necessary. The differentiation between target obstacles and non-target obstacles is often marked by substantial disparities in size, height and volume, with non-target obstacles typically exhibiting small dimensions or volumes.

Therefore, two additional constraints are introduced to identify target obstacles. The first constraint is on the number of points within each cluster, and the second constraint is on the volume and height of the bounding box containing these clusters, with appropriate threshold values set. Subsequently, obstacle clusters that meet these constraints are marked in the detection results using a minimum bounding box. The algorithm pseudo code is shown in Fig. 8.

```
Algorithm: Target Detection and Bounding Box Labeling
Input: Point cloud  nowall_ptcloud
Output: minimum bounding box
Initialize labels array L and numClusters.
Remove invalid points and store them in newptcloud.
Set newLabel .
for each point p in newptcloud do
    if L[p] is not assigned then
        newLabel = newLabel + 1
        assign newLabel to L[p]
        expandCluster(newptcloud, L, p, newLabel, minDistance)
    end if
end for
for each cluster do
    if the number and size of clusters meet the condition
        minimum bounding box(cluster)
    end if
end for
Return minimum bounding box.
```

FIGURE 8. Obstacle detection algorithm.

IV. EXPERIMENTS AND RESULTS

This article employs the Carla simulation platform, widely used in autonomous driving research, to simulate vehicle transportation within a tunnel environment. This simulation serves to validate the effectiveness and detection accuracy of the algorithm. In this setup, a LiDAR sensor is mounted above the vehicle, with the LiDAR's x-direction aligned with the vehicle's forward direction. The experimental scenario is depicted in the illustration below (Fig. 9):

The chosen LiDAR for the experiment is the Horizon model from Livox company. This three-dimensional LiDAR operates at a wavelength of 905 nm and provides a reflectivity of 80% within a range of 260 meters. It boasts a horizontal field of view of 81.7 degrees, a vertical field of view of 25.1 degrees, a distance accuracy of 2cm, and an angular accuracy of less than 0.05 degrees. It outputs a point cloud with a rate of 240,000 points per second.

During the experiment, algorithm parameters were configured based on the characteristics of the mining environment and the point cloud features obtained from the LiDAR scans. The region of interest (ROI) was defined with a length of 50 meters in front of the vehicle, a height of 5 meters, and a width matching the tunnel's width. For downsampling under a voxel grid point cloud, the grid cell size was set to 0.2 meters. During ground point cloud segmentation using an improved RANSAC algorithm, the following parameters were specified: 50 iterations, a distance error threshold of 0.3 meters, and a segment distance threshold of 20 meters. For wall point cloud segmentation, an offset threshold of 1.65 meters was used. During obstacle point cloud clustering, an Euclidean distance threshold of 0.6 meters and a minimum point count of 10 were applied. When filtering obstacles, constraints were imposed on both quantity and size. The quantity constraint ranged from a minimum of 15, the size constraint height is not less than 0.3, the minimum bounding box volume is 0.06, and the maximum is no more than 2.5.

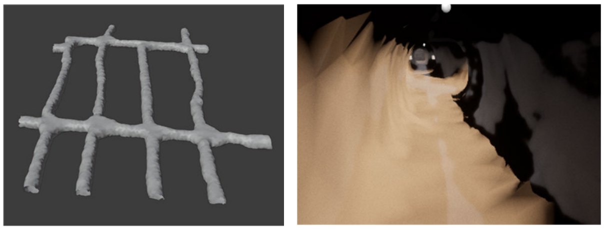


FIGURE 9. **Experimental scenario.**

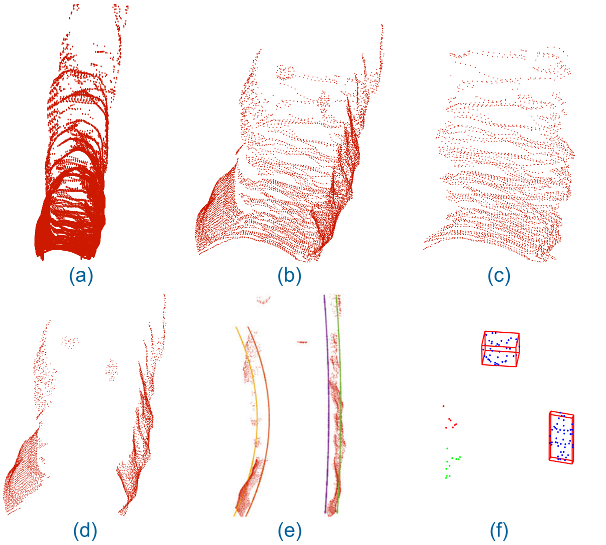
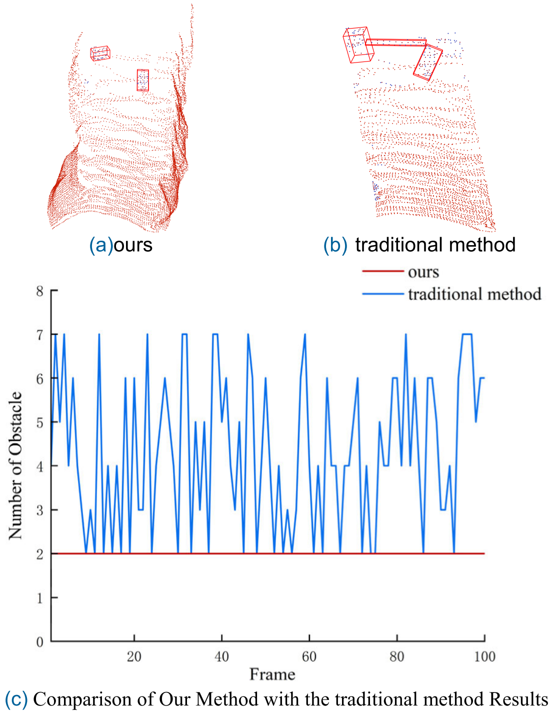


FIGURE 10. **Experimental results of point cloud data processing in underground mine tunnel. (a) The original point cloud dataset. (b) The point cloud after preprocessing. (c) The ground point cloud of the tunnel. (d) The point cloud after ground segmentation. (e) Fitting of tunnel wall curves. (f) Labeling of obstacles.**

Upon this foundation, the validation of the obstacle point cloud recognition method proposed in this manuscript was undertaken. The experimental outcomes, as illustrated in Fig. 10, provided a comprehensive depiction of the process. The initial point cloud data, as depicted in Fig. 10(a), revealed that the initial dataset had been acquired through laser radar scanning, comprising both obstacle point clouds and the surrounding environmental information. Following preprocessing, the resultant point cloud, as seen in Fig. 10(b), demonstrated a significant reduction in data volume, effectively curtailing computational time. Subsequent to ground point cloud segmentation, as displayed in Fig. 10(c), (d), it was evident that, through this phase, the ground point clouds had been predominantly segmented in their entirety. The outcome of the tunnel wall curve fitting, as exemplified in Fig. 10(e), manifested that the majority of wall point clouds lay within the bounds of the fitted curve. Following the removal of wall points, the results of obstacle point cloud labeling, as presented in Fig. 10(f), indicated that wall point clouds had been nearly entirely filtered out, thus enabling the precise identification of obstacles.



(c) Comparison of Our Method with the traditional method Results

FIGURE 11. **The comparison of the processing results of our method and the traditional obstacle detection. In the same environment, Fig. 11(a) is the processing result of our method, Fig. 11(b) is the traditional obstacle detection result, and Fig. 11(c) is the detection result of the two methods when the number of real obstacles is 2 in 100 frames.**

The comparison of the processing results of our method and the traditional obstacle detection method of the road[1] under the same environment is shown in Fig. 11. Fig. 11(a) is the processing result of our method, and Fig. 11(b) is the traditional obstacle detection method result. Compared with the traditional detection method, our proposed method performs better in the tunnel environment. Because our obstacle detection method includes a tunnel wall segmentation module, its detection range is wider than that of road methods, and it performs excellently in various tunnel scenarios, which is also verified in Experiment 1. Fig. 11(c) shows the processing results of the two methods when the number of real obstacles is 2. Our method can accurately detect the obstacles, while the performance of the road detection method in the tunnel environment is unsatisfactory.

To further investigate the performance of the proposed method in underground mines, three sets of experiments were conducted involving different tunnel scenarios, varying

[1] https://github.com/SS47816/lidar_obstacle_detector

numbers of obstacles, and different obstacle distances. These experiments aim to assess the method's versatility and robustness.

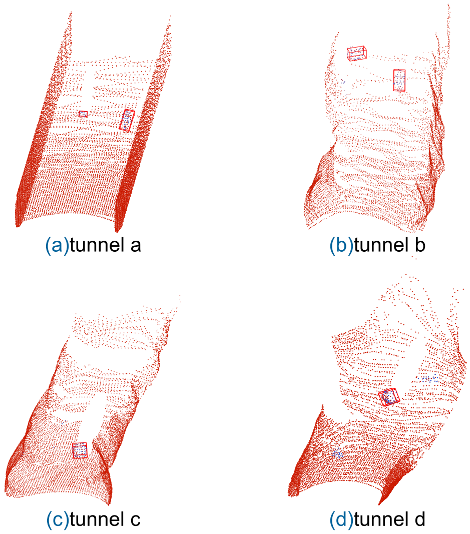

(a)tunnel a
(b)tunnel b
(c)tunnel c
(d)tunnel d

FIGURE 12. Results of tunnel detection for different types of tunnels.

1) EXPERIMENT 1: DIFFERENT TUNNEL TYPES

In this experiment, we meticulously crafted three distinct tunnel environments. As elucidated in Fig 12, Fig 12(a) depicts an engineered tunnel with a regularized structure, while Fig 12(b) and Fig 12(c) provide intricate simulations of authentic tunnel environments. Tunnel detection results of different types of tunnels. Fig 12(a) is a tunnel with a relatively regular road surface, Fig 12(b) and (c) are two different tunnels, and Fig 12(d) is the intersection of two tunnels. The running time statistics of 100 frames of data in different tunnel environments are shown in Fig 13. The results show that the processing time of the proposed method ranges between 0.14 and 1.08 seconds.

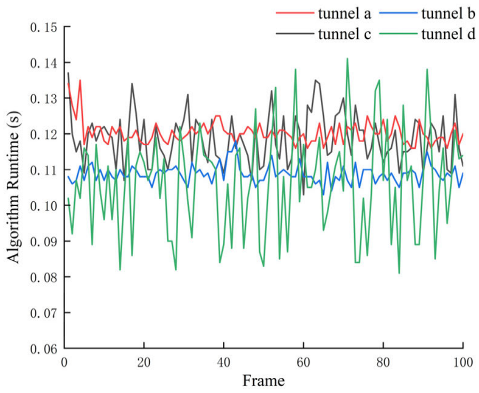


FIGURE 13. Algorithm runtime of 100 frames.

The experimental results unequivocally demonstrate the remarkable adaptability of the proposed method across various tunnel environments. It excels not only in accurately identifying obstacles in straightforward tunnel settings but also exhibits outstanding detection performance in complex tunnel environments. In addition, the processing time of no more than 0.14 s within a 50m range meets real-time detection requirements, suitable for mine trucks with a maximum speed limit of 30 km/h.

TABLE 1. Statistical analysis of detection results for 200 frames at different distances.

Distance	Detection Rate
17 m	97%
35 m	96%
55 m	90%

2) EXPERIMENT 2: DIFFERENT OBSTACLE DISTANCES

The objective of this experiment is to assess the method's performance about the distance between obstacles and the sensor, which is of paramount importance for its practical application. Given that one of the primary safety concerns in underground mine tunnels is the presence of falling rocks, this experiment uses rocks as alternative obstacles for detection. These rocks have an approximate radius of 20 cm and have been strategically placed in front of the vehicle at distances of 17 m, 35 m, and 55 m from the LIDAR.

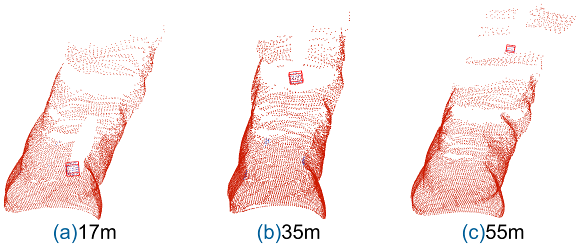

(a)17m
(b)35m
(c)55m

FIGURE 14. Detection results for different distances of obstacle.

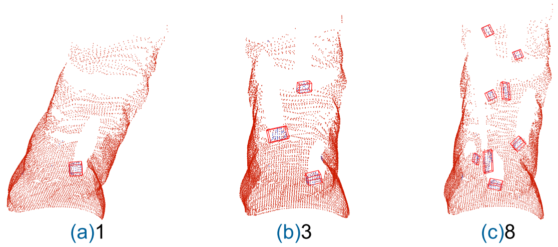

(a)1
(b)3
(c)8

FIGURE 15. Detection results for different numbers of obstacles.

The processing results obtained using the proposed method are shown in Fig. 14. Additionally, the detection rates of 200 frames of data at different distances are recorded in Table 1. The detection rate refers to the probability of correctly detecting obstacles in the tunnel in an experiment.

The statistical results of the detection rate show that when the obstacle is at distances of 17 m and 35 m, the method performs very well with almost no recognition errors. However, when the obstacle is at a distance of

TABLE 2. Statistical analysis of detection results for 200 frames at different distances.

Number	Detection Rate
1	97%
3	99.5%
8	98%

55 m, the detection rate drops significantly to 90%. These results indicate that the accuracy of the method decreases with increased distance because the point cloud becomes sparse when the LiDAR scans far away. Therefore, defining a suitable region of interest is important when using the proposed method.

3) EXPERIMENT 3: DIFFERENT NUMBER OF OBSTACLES

The objective of this experiment is to validate the algorithm's detection performance in environments containing multiple obstacles. The previous experiment provided ample data to establish the detection performance with a single obstacle. In this experiment, we introduce an increased number of obstacles into the tunnel environment, specifically three and eight obstacles, while also diversifying the types of obstacles. To minimize the impact of distance on the detection results, we have confined the obstacles within a 30-meter range.

Fig. 15 shows the experimental results obtained using the proposed method in scenarios with different numbers of obstacles. Additionally, we processed the 200 frames of data collected in these scenarios and summarized the results in Table 2.

The results demonstrate that, in scenarios with 1, 3, and 8 obstacles, the method consistently and accurately identifies the obstacles without any instances of omission. This performance is further supported by the statistical data in Table 2, which highlights the method's outstanding performance across scenarios with varying numbers of obstacles.

Furthermore, the experiment included a diverse range of obstacle types, including rocks and human figures, all of which were accurately identified by the method. This signifies the method's versatility and its suitability for detecting various types of obstacles within the tunnel environment.

However, in scenarios with a higher number of closely spaced obstacles, there were instances where multiple nearby obstacles were identified as a single entity. This may be attributed to the parameter settings within the obstacle distance clustering method. To further optimize the experimental results, it is advisable to experiment with different parameter values through iterative testing. This approach allows for the selection of the most optimal parameter combination, which can lead to improved accuracy in obstacle detection, especially in complex, densely cluttered environments.

V. CONCLUSION

Based on 3D LiDAR, this paper proposes an efficient obstacle detection method for underground mines. This method addresses the limitations of traditional detection methods that perform well on flat roads but struggle in tunnel environments. The method combines the region growing algorithm with RANSAC to segment rugged ground points, significantly improving processing accuracy compared to unimproved methods. Additionally, the innovative tunnel point cloud processing module—encompassing boundary point extraction, fitting, elimination, and the application of Euclidean clustering and obstacle recognition strategies ensures accurate obstacle detection.

Experimental results demonstrate that the proposed method accurately detects obstacles in various tunnel environments, achieving a total detection rate above 95% within a 50-meter region of interest and maintaining a processing time within 0.14 seconds. The method also performs well in environments with different obstacle types and quantities, exhibiting strong robustness and accuracy in underground mines.

It is worth noting that the selection of parameter values directly affects the final results, especially in scenarios with dense obstacles. Iterative testing to select the optimal parameter combination is recommended to achieve better detection results.

REFERENCES

[1] B. Liu, B. Tian, and J. Qiao, "Mine track obstacle detection method based on information fusion," *J. Phys., Conf. Ser.*, vol. 2229, no. 1, Mar. 2022, Art. no. 012023, doi: 10.1088/1742-6596/2229/1/012023.

[2] J. Wang, Y. Xue, J. Xiao, and D. Shi, "Diffusion characteristics of airflow and CO in the dead-end tunnel with different ventilation parameters after tunneling blasting," *ACS Omega*, vol. 8, no. 39, pp. 36269–36283, Sep. 2023, doi: 10.1021/acsomega.3c04819.

[3] Y. Jiang, Z. Li, G. Yang, Y. Zhang, and X. Zhang, "Recent progress on smart mining in China: Unmanned electric locomotive," *Adv. Mech. Eng.*, vol. 9, no. 3, Mar. 2017, Art. no. 168781401769504, doi: 10.1177/1687814017695045.

[4] S. Grehl, M. Donner, M. Ferber, A. Dietze, B. Jung, and H. Mischo, "Mining-RoX-mobile robots in underground mining," in *Proc. 3rd Int. Future Mining Conf.*, Nov. 2015, pp. 4–6.

[5] J. Tang, X. Lu, Y. Ai, B. Tian, and L. Chen, "Road detection for autonomous truck in mine environment," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 839–845, doi: 10.1109/ITSC.2019.8917022.

[6] C. Lu, F. Ji, N. Xiong, S. Jiang, D. Liu, and S. Zhang, "Track obstacle real-time detection of underground electric locomotive based on improved YOLOX," in *Proc. Int. Conf. Smart Comput. Commun.*, 2023, pp. 236–246, doi: 10.1007/978-3-031-28124-2_22.

[7] J. Wang, L. Wang, P. Peng, Y. Jiang, J. Wu, and Y. Liu, "Efficient and accurate mapping method of underground metal mines using mobile mining equipment and solid-state LiDAR," *Measurement*, vol. 221, Nov. 2023, Art. no. 113581, doi: 10.1016/j.measurement.2023.113581.

[8] T. Shi, D. Zhong, and L. Bi, "A new challenge: Detection of small-scale falling rocks on transportation roads in open-pit mines," *Sensors*, vol. 21, no. 10, p. 3548, May 2021, doi: 10.3390/s21103548.

[9] J. Li, M. Qiu, Y. Zhang, N. Xiong, and Z. Li, "A fast obstacle detection method by fusion of double-layer region growing algorithm and grid-SECOND detector," *IEEE Access*, vol. 9, pp. 32053–32063, 2021, doi: 10.1109/ACCESS.2020.3047711.

[10] V.-T. Luu, V.-C. Huynh, V.-H. Tran, T.-H. Nguyen, and T.-N.-H. Phu, "Traditional method meets deep learning in an adaptive lane and obstacle detection system," in *Proc. 5th Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Nov. 2020, pp. 148–152, doi: 10.1109/gtsd50082.2020.9303108.

[11] Z. Tong, W. Zhang, and X. Zhang, "Underground mine road detection using deep learning technique," *Appl. Sci.*, vol. 13, no. 7, p. 4098, Mar. 2023, doi: 10.3390/app13074098.

[12] S. Ruan, S. Li, C. Lu, and Q. Gu, "A real-time negative obstacle detection method for autonomous trucks in open-pit mines," *Sustainability*, vol. 15, no. 1, p. 120, Dec. 2022, doi: 10.3390/su15010120.

[13] R. Zhang, Y. Yang, W. Wang, L. Zeng, J. Chen, and S. McGrath, "An algorithm for obstacle detection based on YOLO and light filed camera," in *Proc. 12th Int. Conf. Sens. Technol. (ICST)*, Dec. 2018, pp. 223–226, doi: 10.1109/ICSensT.2018.8603600.

[14] D. Ball, B. Upcroft, G. Wyeth, P. Corke, A. English, P. Ross, T. Patten, R. Fitch, S. Sukkarieh, and A. Bate, "Vision-based obstacle detection and navigation for an agricultural robot," *J. Field Robot.*, vol. 33, no. 8, pp. 1107–1130, Dec. 2016. Accessed: Oct. 13, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21644

[15] S. Nedevschi, R. Schmidt, R. Danescu, D. Frentiu, T. Marita, T. Graf, F. Oniga, and C. Pocol, "High accuracy stereo vision system for far distance obstacle detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 292–297, doi: 10.1109/IVS.2004.1336397.

[16] F. Gao, C. Li, and B. Zhang, "A dynamic clustering algorithm for LiDAR obstacle detection of autonomous driving system," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25922–25930, Nov. 2021, doi: 10.1109/JSEN.2021.3118365.

[17] A. Moffatt, E. Platt, B. Mondragon, A. Kwok, D. Uryeu, and S. Bhandari, "Obstacle detection and avoidance system for small UAVs using a LiDAR," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 633–640, doi: 10.1109/ICUAS48674.2020.9213897.

[18] Y. Ji, S. Li, C. Peng, H. Xu, R. Cao, and M. Zhang, "Obstacle detection and recognition in farmland based on fusion point cloud data," *Comput. Electron. Agricult.*, vol. 189, Oct. 2021, Art. no. 106409, doi: 10.1016/j.compag.2021.106409.

[19] J. Li, Y. Zhang, X. Liu, X. Zhang, and R. Bai, "Obstacle detection and tracking algorithm based on multi-LiDAR fusion in urban environment," *IET Intell. Transp. Syst.*, vol. 15, no. 11, pp. 1372–1387, Nov. 2021, doi: 10.1049/itr2.12105.

[20] B. Sun, W. Li, H. Liu, J. Yan, S. Gao, and P. Feng, "Obstacle detection of intelligent vehicle based on fusion of LiDAR and machine vision," *Eng. Lett.*, vol. 29, no. 2, pp. 1–9, 2021.

[21] J. Li, R. Li, J. Wang, and M. Yan, "Obstacle information detection method based on multiframe three-dimensional LiDAR point cloud fusion," *Opt. Eng.*, vol. 58, no. 11, p. 1, Nov. 2019, doi: 10.1117/1.oe.58.11.116102.

[22] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3D LiDAR-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robot. Auto. Syst.*, vol. 83, pp. 299–311, Sep. 2016, doi: 10.1016/j.robot.2016.06.007.

[23] Y. Tian, W. Song, S. Fong, S. Zou, E. S. Lee, and J. Rhee, "A 3D obstacle classification method in point clouds using K-NN," in *Proc. 2nd Int. Conf. Big Data Internet Things*, Oct. 2018, pp. 76–79, doi: 10.1145/3289430.3289457.

[24] H. Le and T. Trân, "3D LiDAR segmentation based on Euclidean clustering for embedded system," *TRA Vinh Univ. J. Sci.*, vol. 13, no. 6, Jul. 2023, doi: 10.35382/tvujs.13.6.2023.2120.

[25] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles," *J. Intell. Robotic Syst.*, vol. 76, nos. 3–4, pp. 563–582, Dec. 2014, doi: 10.1007/s10846-013-9889-4.

[26] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3145–3152, Oct. 2018, doi: 10.1109/LRA.2018.2850061.

[27] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3517–3523.

[28] B. Li, "3D fully convolutional network for vehicle detection in point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1513–1518, doi: 10.1109/IROS.2017.8205955.

[29] W. Lyu, W. Ke, H. Sheng, X. Ma, and H. Zhang, "Dynamic downsampling algorithm for 3D point cloud map based on voxel filtering," *Appl. Sci.*, vol. 14, no. 8, p. 3160, Apr. 2024, doi: 10.3390/app14083160.

[30] Y. Liu, Z. Yang, J. Tong, J. Yang, J. Peng, L. Zhang, and W. Cheng, "ET-PointPillars: Improved PointPillars for 3D object detection based on optimized voxel downsampling," *Mach. Vis. Appl.*, vol. 35, no. 3, p. 56, Apr. 2024, doi: 10.1007/s00138-024-01538-y.

[31] J. Huang, Z. Zheng, Y. Zhang, and Y. Zhou, "Segmentation clustering algorithm for point cloud processing of targets on the ground," *J. Phys., Conf. Ser.*, vol. 2478, no. 6, Jun. 2023, Art. no. 062023, doi: 10.1088/1742-6596/2478/6/062023.

PINGAN PENG received the Ph.D. degree in mining engineering from Central South University, in 2019. He is currently an Associate Professor with Central South University. His current research interests include intelligent mining, unmanned driving, and microseismic monitoring. He has presided over the development of the first mine microseismic monitoring system in the field of metal mines with a fully independent property rights and an automatic driving system for trackless equipment in underground mines, which have been popularized and applied in mines on a large scale.

JIN PAN received the B.S. degree in safety engineering from Xiangtan University, China, in 2022. She is currently pursuing the M.S. degree with Central South University. Her current research interests include obstacle detection for unmanned vehicles in smart mines and autonomous exploration for quadruped robots.

ZIYU ZHAO received the master's degree in mining engineering from Central South University, Changsha, China, in 2021, where he is currently pursuing the Ph.D. degree in mining engineering. His research interests include digital mining and intelligent mining.

MENGNAN XI received the B.S. degree in safety engineering from China University of Petroleum (East China), in 2022. She is currently pursuing the M.S. degree with Central South University. Her current research interests include scanning and modeling hazardous areas in mines using quadruped robots.

LINXINGZI CHEN received the B.S. degree in safety engineering from Jiangxi University of Science and Technology, China, in 2022. She is currently pursuing the M.S. degree with Central South University. Her current research interests include object detection and image recognition based on deep learning, as well as automatic inspection by quadrupled robots in metal mines.