

Berm Detection for Autonomous truck in Surface Mine Dump Area

Dejiang Meng¹, Bin Tian², Ziyu Pan³, Dongpu Cao⁴, Long Chen⁵

Abstract— To ensure an autonomous truck can operate safely in a dump area, it is crucial to detect a berm accurately in advance. However, there are two challenges. First, the berm is not a static terrain but a movable one because of soil dumping. Second, berms are often irregular in shape—they are neither straight lines nor smooth curves. We considered two types of possible existing methods, but only to find they are not accurate and can't provide height information. Therefore, this paper proposes a berm detection algorithm, which includes three steps. First, extract berm candidate 3D LiDAR points based on a 2D height difference grid map. Second, use a binary Bayes filter to build and update 3D dynamic probability grid maps. Last, use a fitting rectangle technique to recognize the berm. We call this algorithm a Probability Grid Berm Detection (PGBD) algorithm. Off-line experimental evaluations on PGBD carried on datasets show good performance, compared with two curb detection algorithms, which are Hough Transformation and Haar Wavelet Transformation. And the good performance of the PGBD algorithm is further verified in the real-time experiment.

I. INTRODUCTION

Berm is used to avoid trucks going over the edge at a dump point. As shown in Fig. 1, if trucks go over the edge, it will be dangerous. A berm is shown in Fig. 2(b). Note that it is not a static terrain but a dynamic one because of constant soil dumping.

For autonomous trucks, a berm cannot be detected as an obstacle. The reason is that if it is a berm, a truck needs back itself to the berm at a close distance for soil dumping, and if it is an obstacle, the truck has to avoid it at a relatively far distance. So, to ensure autonomous trucks operate safely in dump areas, it is crucial to detect berms in advance for path planning and speed planning.

There are some challenges in the methods currently available for berm detection. One of the methods is to utilize high-definition map (HD map). HD map is usually applied to static terrain and gets updated with low frequency, but the fast

change of berm's position requires map updating in real-time, that means perceiving berms is out of HD map's capacity. The second type of possible methods that we have considered refers to those applied to detect curbs. The reason we consider to borrow curb perception methods is that curb has a similar feature to berm: points on both curb and berm are higher than those on the ground near them. However, based on our experiments, curb perception methods don't work well either in detecting berm, for the reason that berm is irregular, unlike curb, which is regular. Another main reason for not using curb perception methods lies with the failure for curb algorithms to provide height information, which is important because it determines whether the truck can back itself to the berm for soil dumping. The height standard varies from mine to mine. If the height is within the standard, the autonomous truck will be allowed to back itself for soil dumping, otherwise it will not.

In response to the current challenges, this paper proposes a berm detection algorithm, which includes three steps. First, extract berm candidate 3D LiDAR points based on a 2D height difference grid map. Second, use a binary Bayes filter to build and update 3D dynamic probability grid maps. Last, use a fitting rectangle technique to recognize the berm.

The main contribution of this paper is the proposed PGBD algorithm. Integrating 2D grid map with 3D dynamic probability grid map, the algorithm is able to minimize the negative effect of berm's dynamism and irregularities on the detection accuracy. Besides, it enables an accurate output of the berm's present height.

This method has a limitation—the way to differentiate a berm and an obstacle only applies to dump areas where a berm's length exceeds 8m and the length of all dynamic obstacles (other trucks and pedestrians) is less than 8m.



Fig. 1. Trucks went over the edge at dump points.

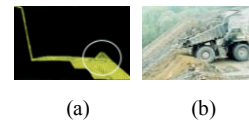


Fig. 2. (a) is the diagrammatic drawing of the dump area, where the berm is inside the white circle; (b) is a berm in a surface mine.

This paper is structured as follows. Following the discussion of the researches on related work in section II, we propose a berm detection algorithm in section III. To verify its performance, we demonstrate both online and offline experiment processes and results in section IV. Finally, section V discusses its contributions and limitations.

*This work was supported by the Key-Area Research and Development Program of Guangdong Province (2020B090921003, 2020B0909050001), the National Natural Science Foundation of China (61503380, 61773381).

¹D. Meng is with the Beijing Institute of Technology and with the Waytous Inc., Beijing 100190, China (e-mail: 1149834447@qq.com).

²B. Tian is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, and with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (corresponding author; e-mail: bin.tian@ia.ac.cn).

³Z. Pan is with the School of Data and Computer Science, Sun Yat-sen University and with the Waytous Inc., China (e-mail: ziyu.pan@waytous.com).

⁴D. Cao is with Department of Mechanical and Mechatronics Engineering, University of Waterloo, Ontario N2L3G1, Canada. (email: dongpu@uwaterloo.ca).

⁵L. Chen is with the School of Data and Computer Science, Sun Yat-sen University, China (e-mail: chen146@mail.sysu.edu.cn).

II. RELATED WORK

We used Lanelet2 [1] to build a HD map of a dump area. The result shows that the position accuracy of a berm is poor, because HD map is usually applied to static terrain and gets updated with low frequency, but the fast change of berm's position requires map updating in real-time, out of HD map's capacity. However, the method we propose is real-time, so the detection accuracy will not be affected by a berm's changing position.

We also conducted research on algorithms often used for curb detection in this section for its similarity of shape to a berm. Curb detection algorithms can be divided into two categories: camera-based and LiDAR-based algorithms. In camera-based algorithms, the main sensor is camera, and in LiDAR-based algorithms, the main sensor is LiDAR.

Camera-based algorithms detect road curbs by designing a novel network [2], [3], [4], or using traditional engineering techniques, such as Hough Transformation, Kalman Filter, and RANSAC [5], [6], [7]. These methods can neither be applied at night because of their requirement for daylight, nor can provide a berm's height. However, our method can achieve both, meaning it is not subject to the lack of daylight, and also can provide height information.

Some LiDAR-based algorithms include three key steps: first, integrate a sequence of LiDAR point clouds and project them onto a bird's-eye view image; second, use this image to detect the curb with convolutional neural networks or convolutional recurrent network; last, filter out noise through fusing detections in subsequent images and through tracking detected curbs over time [8], [9]. Others use traditional engineering techniques such as Haar Wavelet Transformation, Hough Transformation, Smooth Arc Length Feature, and Condition Random Field (CRF) [10], [11], [12], [13]. These methods fail to provide a berm's height or to consider its irregular shape.

Aimed at the existing challenges of these methods can be used for berm detection, we propose a new berm detection algorithm. This algorithm first obtains a 2D height difference grid map to extract berm candidate 3D LiDAR points. The 2D height difference grid map not only contains the coordinate of the two-dimensional cell but also includes the height difference of the LiDAR point clouds in a two-dimensional cell. Second, it builds a 3D dynamic probability grid map based on the berm candidate 3D LiDAR points and uses a binary Bayes filter [14] to update the map's probability. Last, with the 3D dynamic probability grid map, we use the fitting rectangle technique [15] and the skeleton extraction technique [16] to extract a 2D berm grid map. Coordinate and value of a 2D cell occupied by a berm is the position and the maximum height of the berm in the 2D cell respectively. Therefore, we call this algorithm the Probability Grid Berm Detection (PGBD) algorithm.

There is one problem, which is that on the 2D grid map we use, the representation of a berm and an obstacle on the map could be mistaken as the same when they are perceived and processed as the smallest fitting rectangle (The rectangle encloses the occupied cells of a 2D occupancy grid map, and is perpendicular to the truck on the map. The 2D occupancy grid map is introduced in section III.C). Therefore, to prevent

obstacles such as other trucks and pedestrians from being detected as a berm, we have to find the difference in features of a berm and an obstacle. And the difference we identified is in their length disparity.

We have investigated some surface mines and found that a truck is usually the longest obstacle in a dump area, and its length is less than 8m. Although berms vary in length, it is usually longer than 8m. Thereby, we assume that the length of the obstacles in a dump area is less than 8m, and the length of the berm is greater than 8m. So, we define the rectangle on the map, whose length is greater than 8m, as representation of berms.

III. BERM DETECTION

Before detecting the berm, it is necessary to calibrate the raw point clouds and correct their distortion.

This algorithm fuses multi-sensor information from LiDAR, GPS-RTK, and INS. Because LiDAR can provide accurate 3D position information of the environment; GPS-RTK and INS can provide accurate pose information of the autonomous truck. And multi-sensor information needs a unified coordinate system to improve the accuracy of the algorithm. To do so, we calibrate multi-sensor information to a vehicle coordinate system using the method of [17]. The calibrated point cloud position accuracy is 0.02m. As shown in Fig. 3, the definition of the vehicle coordinate system is as follows: its origin coincides with projection of the center of the rear axis of the truck onto the ground; the positive x-axis (the red line) points to the front of the vehicle; the positive y-axis (the green line) to the left of it; and the positive z-axis (the blue line) is the vertical axis pointing upwards. So, this coordinate system doesn't depend on the suspension.



Fig. 3. The left side is CMT96; the upper right corner is RS-Bpearl; the lower right corner is OXTS Inertial+. And the origin of the vehicle coordinate system coincides with projection of the center of the rear axis onto the ground; the red line is x-axis; the green line is y-axis; and the blue line is z-axis.

Mechanical scanning LiDAR will not cause point clouds distortion due to movement. If the LiDAR is a mechanical scanning LiDAR, we use the method of [18] to correct the distorted point clouds.

A. Extraction of Berm Candidate 3D LiDAR Points

Berm candidate 3D LiDAR points refer to LiDAR raw point clouds that hit the berm and obstacles. There are three steps to extract them.

First, we define a frame of point clouds in the vehicle coordinate system as $P^V = \{P_0^V, \dots, P_i^V, \dots, P_n^V\}$, $P_i^V = \{X_i^V, Y_i^V, Z_i^V\}$. And we project the P^V onto a 2D grid map to obtain a 2D height difference grid map (Fig. 4).

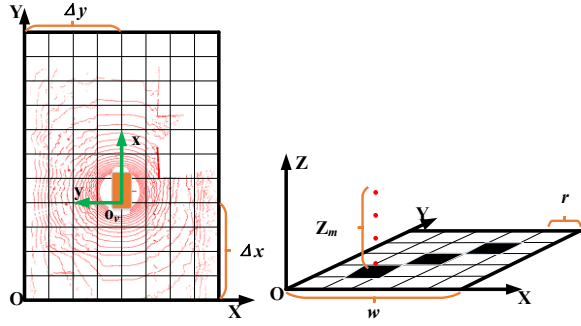


Fig. 4. 2D height difference grid map.

During the projection, we apply the following equation to calculate the height difference.

$$Z_m = \max \{Z_i^V\} - \min \{Z_i^V\} \quad (1)$$

Where Z_m is the height difference of point clouds included in the two-dimensional cell whose index No. is m ; the m is generated by the following equation (2):

$$m = \left\lfloor \frac{X_i^V + \Delta x}{r} + \frac{1}{2} \right\rfloor w + \left\lfloor \frac{-Y_i^V + \Delta y}{r} + \frac{1}{2} \right\rfloor \quad i \in [0, n] \quad (2)$$

Where $\lfloor \cdot \rfloor$ is the rounding down operator; Δx and Δy are the vertical and horizontal offsets between the vehicle coordinate system center and the 2D grid map coordinate system center; r is the resolution of the map; w represents the number of cells along the X-axis direction; the unit of Δx , Δy and r is meter.

Second, before obtaining berm candidate 3D LiDAR points, we need to acquire berm candidate 2D points, which are positions of cells greater than a height difference threshold in the 2D height difference grid map. The threshold is set according to the height standard of a berm.

Last, raw 3D LiDAR points that are projected onto berm candidate 2D points are target points we look for.

As shown in Fig. 5, the coordinate system refers to the vehicle coordinate system; the green sparse points to the raw 3D LiDAR points; and the white sparse points in the yellow rectangle to the berm candidate 3D LiDAR points.

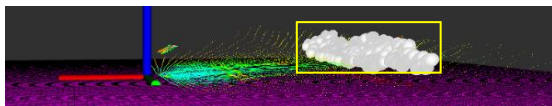


Fig. 5. The coordinate system refers to the vehicle coordinate system; the green sparse points to the raw 3D LiDAR points; and the white sparse points in the yellow rectangle to the candidate 3D LiDAR points of the berm.

B. Build and Update of 3D Dynamic Probability Grid Map

3D dynamic probability grid map can output the confidence of the berm candidate 3D LiDAR points. We need to convert coordinates of the berm candidate 3D LiDAR points from the vehicle coordinate system to the world coordinate system using vehicle's pose, before building and updating the map.

There are some characteristics of the 3D dynamic grid map: the pose of the map's center is equal to the pose of the

vehicle of the first frame that the map contains; coordinates of the berm candidate 3D LiDAR points need to be converted from the world coordinate system to the map coordinate system; the value of the cell of the map represents occupancy probability; every map only contains a fixed number of frames of berm candidate 3D LiDAR points and the map's size is adaptive.

Because the size of the map is adaptive, so, it is managed by using an octree [19] whose spatial range can be constantly expanded. Octree is a tree data structure that can reduce the time complexity and space complexity of our code.

There are four steps to build an octree (or a map) using the frame of berm candidate 3D LiDAR points we have created:

- 1) Set the resolution of the deepest node, the initial size, and the position of the root node.
- 2) Input every point of the frame of berm candidate 3D LiDAR points into the octree.
- 3) Calculate whether the point is beyond the range defined by the root node; if so, a larger range node should be extended to be the upper layer of the current root node as the new root node. The range of the root node should be repeatedly verified and the root node should be extended until the point is contained in the range of the root node.
- 4) Index the node layer by layer until maximum recursion depth (or minimum volume) based on the coordinate of the point. When an empty node does not reach the maximum recursion depth, a new space will be opened up, and the cube voxel represented by this node will be divided into eight equal parts so as to establish eight new child nodes. And then, we repeat the indexing process until the node reaches its maximum recursion depth (or minimum volume). After that, we insert the point into the node. In order to use a binary Bayes filter [14], we stipulate that a node only be inserted once by a frame of berm candidate 3D LiDAR points.

We use a binary Bayes filter to update the probability of the 3D dynamic grid map. The filter can be used to estimate the probability of Binary questions. Every time after a frame of berm candidate 3D LiDAR points is inserted into the map, we update the map's probability as follows:

$$p(m_i = 1 | z_{1:n}) = \frac{e^S}{1 + e^S} \quad (3)$$

$$S = \log \left(\frac{p(m_i = 1 | z_{1:n-1})}{1 - p(m_i = 1 | z_{1:n-1})} \frac{p(m_i = 1 | z_n)}{1 - p(m_i = 1 | z_n)} \right) \quad (4)$$

$$p(m_i = 1 | z_n) = C \quad (5)$$

Where $p(m_i = 1 | z_{1:n})$ represents the occupancy probability of the cell whose index No. is i , after berm candidate 3D LiDAR points hit the cell for n times; $p(m_i = 1 | z_{1:n-1})$ represents the occupancy probability of the cell, after berm candidate 3D LiDAR points hit the cell for $n-1$ times; $p(m_i = 1 | z_n) = C$ represents the occupancy probability (a constant) of the cell, the probability is only estimated by a point in the berm candidate 3D LiDAR points, which is the point when the cell is hit for the n th time.

To reduce space complexity of a 3D dynamic grid map. After a map has been inserted berm candidate 3D LiDAR points of a fixed number of frames, the map will be marked as finished and be discarded. To ensure the probabilities of the cells in adjacent maps corresponding to the same berm candidate 3D LiDAR point are close, we stipulate that the next map starts to be built when a frame of berm candidate 3D LiDAR points of current map is inserted, and this frame is the 10th from the last frame. So, there are at most two unfinished maps at any time and we use the unfinished map with the most inserts to extract a 2D berm grid map.

C. Extraction of 2D Berm Grid Map

Because Cells of 2D berm grid map is discrete, it can be used to represent an irregular berm, so as to improve the detection accuracy. At the same time, the value of the cell represents the height of the berm, which solves the problem that current research methods cannot provide the height information. There are five steps to extract a 2D berm grid map.

First, we set a probability threshold and a berm length threshold. And, we find 3D probability cells of the 3D dynamic grid map which must be greater than the probability threshold.

Second, we convert coordinates of the 3D probability cells from the 3D dynamic grid map coordinate system to the Vehicle coordinate system of the current frame.

Third, we project the 3D probability cells of the Vehicle coordinate system onto a 2D grid map to acquire a 2D occupancy grid map. Value of projected cells is 1 and value of other cells is 0. The 2D occupancy grid map not only contains the coordinate of the two-dimensional cell but also includes the occupied state (0 or 1).

Fourth, we use the fitting rectangle technique [17] to fit the occupancy area of the 2D occupancy grid map. The result is a rectangle, which is the smallest rectangle enclosing the occupied cells and is perpendicular to the truck. If the length of the fitted rectangle is greater than the berm length threshold, we use the skeleton extraction technique [18] to extract skeleton cells of the occupied area inside the rectangle and set the value of all other cells to 0.

Last, reset the value of a skeleton cell to the maximum height value of the 3D probability cells projected to the skeleton cell and we will obtain the 2D berm grid map.

As shown in Fig. 6, this is a cropped figure of a 2D berm grid map, and the berm detection result is in the yellow rectangle. The value of cells in the red area is 0, which is the non-berm area; the height or color of the 2D cell represents the height of the berm; and the coordinate of the occupied cell is the berm's position.

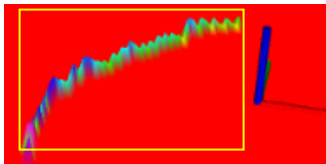


Fig. 6. A cropped figure of a 2D berm grid map. The berm detection result is in the yellow rectangle. The value of cells in the red area is 0, which is the non-berm area; the height or color of the 2D cell represents the height of the berm; and the coordinate of the occupied cell is the berm's position.

IV. EXPERIMENT RESULTS AND ANALYSIS

A. Experiment Scenes

We have done experiments in two different surface mines. Fig. 7(a) is the Baoli surface coal mine in Inner Mongolia, China. The standard height range of the berm in the Baoli surface coal mine is between 0.6m and 1m. Fig. 7(b) is the Heshangqiao surface iron mine in Anhui, China. The standard height range of the berm in the Heshangqiao surface iron mine is between 0.4 meters and 0.8 meters. The width of the berm is usually less than 0.5 meters. We select midpoints in the width direction of the berm as the labeled points. As shown in Fig. 7, berm points in Google Earth are labeled in red.

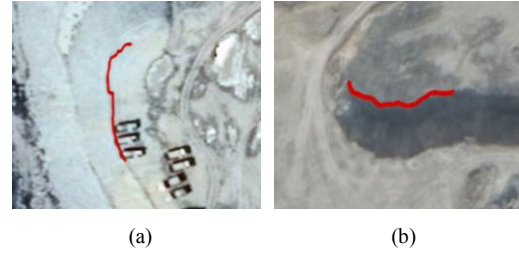


Fig. 7. Berm points in Google Earth are labeled in red. (a) is the Baoli surface coal mine in Inner Mongolia, China. (b) is the Heshangqiao surface iron mine in Anhui, China.

B. Experimental Platform

The experimental platform that this paper adopts is shown in Fig.3. We use the same experimental platform to do experiments in Baoli surface coal mine and Heshangqiao surface iron mine, so as to ensure the experiment results are only affected by the different scenes. The experimental platform's model is CMT96 of Shandong Lingong Construction Machinery Co., Ltd. (SDLG). A 3D LiDAR whose model is RS-Bpearl was mounted on the rear of the autonomous truck and the height is 1.2 meters. As shown in Table I, the ranging accuracy of the LiDAR is very high and the frequency is 10hz. So, the frequency of this algorithm is also 10hz. Similarly, an INS whose model is OXTS Inertial+ was mounted on the rear of the autonomous truck and the height is 1.5 meters. Performance parameters of the INS are shown in Table II.

TABLE I. PERFORMANCE PARAMETERS OF RS-BPEARL

| Parameters | Value |
|-----------------------|----------|
| Range | 0.1m-30m |
| Range Accuracy | ±3cm |
| Horizontal FoV | 360° |
| Vertical FoV | 90° |
| Horizontal Resolution | 0.2° |
| Vertical Resolution | 2.81° |
| Frame Rate | 10Hz |

TABLE II. PERFORMANCE PARAMETERS OF OXTS INERTIAL+

| Parameters | Value |
|---------------------|-------|
| Position Accuracy | 0.02m |
| Roll/Pitch Accuracy | 0.03° |
| Heading Accuracy | 0.1° |
| Frame Rate | 100Hz |

We use an Industrial Personal Computer (IPC) whose model is MIC-770 to run the program. The IPC is a high-performance intel core i7-6825EQ processor-based controller. With the 2.8 GHz base frequency, the IPC can

process point clouds data in real-time. The operating system is Ubuntu18.04.

There are some performance indexes of the 3D dynamic probability grid map we defined: every 3D dynamic probability grid map contains at most 60 frames of berm candidate 3D LiDAR points; the resolution is 0.1 meters. And there are some performance indexes of the 2D grid map we defined: the size of the grid map is $40 \times 40 \text{m}^2$; the resolution is 0.1 meters; the origin is at the lower left corner; the coordinate of the origin of the Vehicle coordinate system is (20,20), and the unit is meter.

C. Results in Dataset

We compared the PGBD algorithm with two state-of-the-art curb detection algorithms which are Hough Transformation and Haar Wavelet Transformation to evaluate their performance [12], [13]. To facilitate comparison of data, the experiments were carried out in MATLAB using offline datasets, which were collected by the experimental platform in the Baoli surface coal mine. We have done experiments using the four datasets collected in the process of backing the autonomous truck at four different areas in Fig. 7(a). The truck velocity is 10 km / h and every dataset includes 200 frames of data. The datasets consist of raw point clouds from LiDAR, poses of the INS, and the reference data sequences (the coordinates of the labeled data of Fig. 7(a) in the Vehicle coordinate system of each frame).

The detection results of one frame in each dataset of the three algorithms are shown in Fig. 8. The results in Fig. 8(a) and Fig. 8(b) show that the Hough Transformation algorithm has many points far away from the reference values, so it is difficult for the Hough Transformation algorithm to handle an irregular-shaped berm. From Fig. 8(c) and Fig. 8(d), we can find that the Haar Wavelet Transformation algorithm has many false-positive results and it cannot be adopted in the dump area without noise filtering. As shown in Fig. 8(a), Fig. 8(b), Fig. 8(c) and Fig. 8(d), the results of the PGBD algorithm we propose are relatively accurate in the irregular-shaped berm, and there are few false-positive results.

There are some performance indexes of the three algorithms in Table III. To evaluate accuracy and average period of the PGBD algorithm and the Hough Transformation algorithm, we selected a group of points at intervals of 0.1 meters in the direction of the berm in Fig. 8(a). To evaluate accuracy and average period of the Haar Wavelet Transformation algorithm, we also selected a group of points at intervals of 0.1 meters in the direction of the berm in Fig. 8(c). According to the actual situation, the width of the berm is usually less than 0.5 meters. And the midpoints in the width direction of the berm are the labeled points. So, if the difference between the detection result and the true value is less than 0.25 meters, we determine that the detection result is accurate. Though the Haar Wavelet Transformation algorithm has a short average period, its accuracy is lower than the PGBD algorithm. The accuracy of the Hough Transformation algorithm is relatively high, but its average period is the longest. The PGBD algorithm not only has the highest accuracy but also its time complexity is lower. Since the Haar Wavelet Transformation algorithm and Hough

Transformation algorithm both have obvious disadvantages, we only test the PGBD algorithm in the real-time experiment.

TABLE III. PERFORMANCE INDEX OF THREE ALGORITHMS

| Algorithms | Average period | Accuracy |
|------------|----------------|----------|
| PGBD | 63.2ms | 96.1% |
| Hough | 82.6ms | 91.3% |
| Wavelet | 19.8ms | 74.2% |

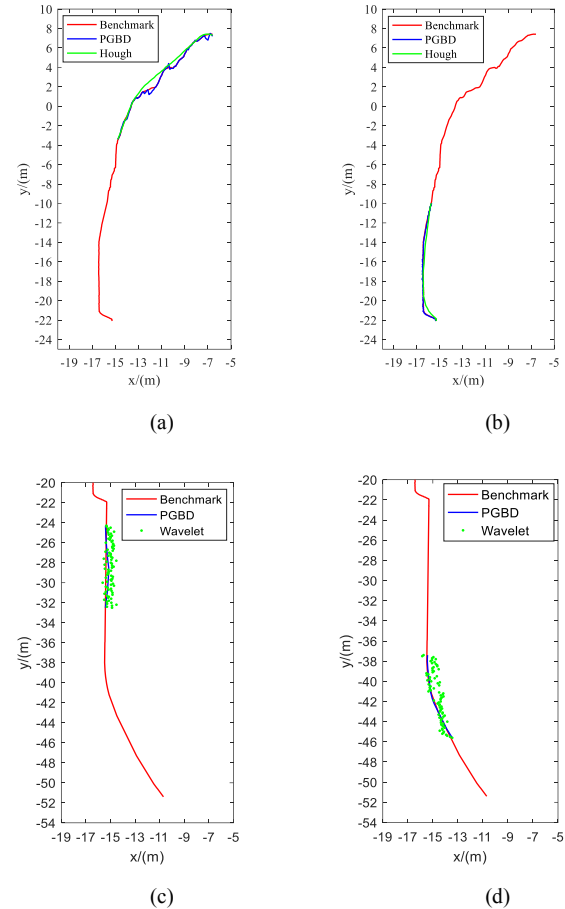


Fig. 8. The detection results of one frame in each dataset of the three algorithms.

D. Real-time Experiment

After the comparison of the three algorithms, we tested the PGBD algorithm using the experimental platform in Baoli surface coal mine and Heshangqiao surface iron mine. Experimental results in Baoli surface coal mine are shown in Fig. 9(a) and Fig. 9(b). And experimental results in Heshangqiao surface iron mine are shown in Fig. 9(c) and Fig. 9(d). The left half of each figure is the berm 3D probability points which is greater than the probability threshold, and the right half of each figure is the 2D berm grid map. Different color of the berm 3D probability points represents different probability, and blue represents the maximum probability. As we can see from Fig. 10, the 2D berm grid map can provide height information of the berm. Besides, through the real-time experiment, it can be concluded that the farthest detection distance of the algorithm is 15 meters.

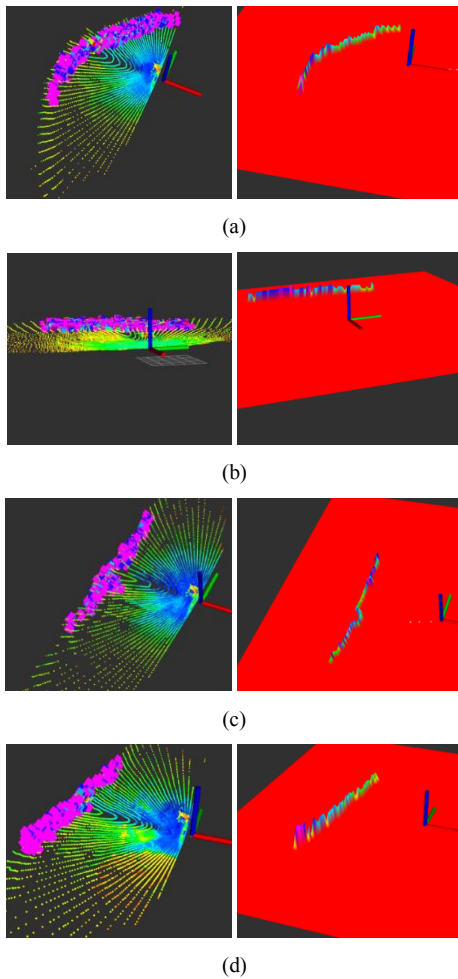


Fig. 9. Real-time detection results: the left half of each figure is the berm 3D probability points which is greater than the probability threshold; different color of the berm 3D probability points represents different probability, and blue represents the maximum probability; the right half of each figure is the 2D berm grid map.

V. CONCLUSION

This paper proposes a berm detection algorithm for the safety of trucks. The main contributions of the PGBD algorithm reside with its ability to provide berm height and accurate detection of irregular-shape berms. And the algorithm's two advantages have been proved by experiment results.

There are two limitations for our research. The first one is that our algorithm can only work within 15meters. So, future researchers can try to expend detection distance. Another one is the limited numbers of dump areas where it can apply. This algorithm targets dump areas where a berm is longer than 8m and all other dynamic obstacles shorter than 8m. And future work can focus on finding out methods that can be applied to a larger scale of different dump areas.

REFERENCES

- [1] F. Poggenghans, et al., "Lanelet2: A high-definition map framework for the future of automated driving," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, pp. 1672-1679, 2018.
- [2] Z. Xu, Y. Sun and M. Liu, "iCurb: Imitation Learning-Based Detection of Road Curbs Using Aerial Images for Autonomous Driving," in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1097-1104, April 2021.
- [3] S. E. C. Goga and S. Nedeveschi, "Fusing semantic labeled camera images and 3D LiDAR data for the detection of urban curbs," *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, pp. 301-308, 2018.
- [4] L. Chen, W. Zhan, W. Tian, Y. He and Q. Zou, "Deep Integration: A Multi-Label Architecture for Road Scene Recognition," in *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4883-4898, Oct. 2019.
- [5] M. Cheng, Y. Zhang, Y. Su, J. M. Alvarez and H. Kong, "Curb Detection for Road and Sidewalk Detection," in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10330-10342, Nov. 2018.
- [6] F. Oniga, S. Nedeveschi and M. M. Meinecke, "Curb Detection Based on a Multi-Frame Persistence Map for Urban Driving Scenarios," *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Beijing, China, pp. 67-72, 2008.
- [7] I. Baek, et al., "CurbScan: Curb Detection and Tracking Using Multi-Sensor Fusion," *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, pp. 1-8, 2020.
- [8] T. Suleymanov, L. Kunze and P. Newman, "Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, pp. 2693-2700, 2019.
- [9] J. Liang, N. Homayounfar, W. Ma, S. Wang and R. Urtasun, "Convolutional Recurrent Network for Road Boundary Extraction," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 9504-9513, 2019.
- [10] Y. Zhang, J. Wang, X. Wang, C. Li and L. Wang, "3D LIDAR-Based Intersection Recognition and Road Boundary Detection Method for Unmanned Ground Vehicle," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Gran Canaria, Spain, pp. 499-504, 2015.
- [11] S. E. Catalina Deac, I. Giosan and S. Nedeveschi, "Curb detection in urban traffic scenarios using LiDARs point cloud and semantically segmented color images," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, pp. 3433-3440, 2019.
- [12] S. Xu, R. Wang and H. Zheng, "Road Curb Extraction from Mobile LiDAR Point Clouds," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 996-1009, Feb. 2017.
- [13] G. Wang, J. Wu, R. He and S. Yang, "A Point Cloud-Based Robust Road Curb Detection and Tracking Method," in *IEEE Access*, vol. 7, pp. 24611-24625, 2019.
- [14] S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*. Boston: MIT-Verlag, 2005.
- [15] S. Satoshi, "Topological structural analysis of digitized binary images by border following." In *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, 1985.
- [16] A. Waleed, "Skeletonization Algorithm for Binary Images." In *Procedia Technology*, vol. 11, 2013.
- [17] H. Kim, S. N. Kasturi Rangan, S. Pagad and V. G. Yalla, "Motion-based Calibration between Multiple LiDARs and INS with Rigid Body Constraint on Vehicle Platform," *2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA, pp. 2058-2064, 2020.
- [18] T. Renzler, M. Stolz, M. Schratter and D. Watzenig, "Increased Accuracy for Fast Moving LiDARS: Correction of Distorted Point Clouds," *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Dubrovnik, Croatia, pp. 1-6, 2020.
- [19] H. Armin, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," in *Autonomous Robots*, vol. 34, no. 3, 2013.