3D-Lidar based obstacle detection and fast map reconstruction in rough terrain*

Ning Li
NORINCO Unmanned Vehicle Research and Development Center
China North Vehicle Research Institute
Beijing, China
lining199003@163.com

Bo Su
NORINCO Unmanned Vehicle Research and Development Center
China North Vehicle Research Institute
Beijing, China
bosu@noveri.com.cn

***Abstract*—Obstacle detection based on 3D-Lidar is an important sensing means of automation navigation. In rough terrain, due to the unstructured characteristics of the field environment and the bumpy road, it is difficult to accurately detect obstacles by Lidar in a single scan. Meanwhile the reconstruction of obstacles in the blind area of the sensor is also a key problem of autonomous navigation for safety. We propose a 3D-Lidar based obstacle detection and fast map reconstruction method to solve these problems in rough terrain. First, point clouds are projected onto the 2D grid map of overhead view. we use gradient statistics in the height direction to obtain a basic obstacle detection result. Then according to the obstacle detection result in single frame, the proposed method uses Bayesian probability calculation based on the first-order Markov model for perception map reconstruction. The final perception result consists of the current frame perception result and historical results by map reconstruction. And a probability map on the confidence value of obstacle is generated rather than the traditional binary grid map of obstacle and obstacle-free area. This method has high robustness for obstacle detection in rough terrain and can effectively solve the problem of environmental perception in unstructured scenes. In addition, obstacle in the blind area of the sensor is retained by map reconstruction to ensure the safe driving of the UGV. The calculation period of the method is within 100ms and meets the real-time requirement for autonomous navigation of the UGV.**

Keywords—3D-Lidar, Obstacle detection, Map reconstruction, UGV

I. INTRODUCTION

At present, 3D-Lidar has become an indispensable sensing sensor in the field of robot research, especially in the field of autonomous navigation of unmanned ground vehicle (UGV) [1-3]. 3D-Lidar based environmental perception play an important part in obstacle detection [4], simultaneous localization and mapping [5], semantic segmentation [6], target detection [7] and so on. This paper focuses on solving the problem of obstacle detection for the unmanned ground vehicle (UGV) in rough terrain in order to ensure the safety in autonomous navigation. And obstacle reconstruction in the blind area is regard as an important research content under consideration.

* Research supported by the National Natural Science Foundation of China (Grant No. 91748211).

A. Related work

As 3D-Lidar is increasingly used in the field of self-driving, there are some algorithms applied to obstacle detection. But most of these algorithms are not specifically applied to rough terrain.

Road plane segmentation based algorithm is widely used in obstacle detection in structured scenes of urban area. In paper [8], first point clouds are projected onto the overhead view. Then the road boundary detection is used to identify the road plane area in the overhead view, and the Random Sample Consensus (RANSAC) algorithm is applied to solve the road plane equation. Finally, for point clouds that do not meet the road plane equation, they are considered as obstacle point clouds. This method is very effective for smooth roads such as expressways and urban roads. However, for rough terrain such as the bumpy road surface, as it is difficult to fit the plane equation, the method is not a good solution in this scene.

Other researchers try to use deep learning to solve obstacle detection. Using convolution neural network for scene segmentation to detect obstacle is another method [9]. It has a good result for vehicle and pedestrian detection. Due to the irregular characteristics of most scattered obstacles such as gravel blocks, it is difficult to perform effective obstacle labeling and feature extraction for training. Meanwhile the generalization ability of deep learning based obstacle detection is weak, so it is easy to cause over-fit for unknown scenes.

Paper [10] has made the research on environmental perception for UGV in unstructured scenes. Based on the sensor model of HDL-64 3D-Lidar, the obstacle is detected by calculating the gradient height. As the method is used for autonomous navigation of micro-UGV with low speed, there is no special research on obstacle detection in the blind area which is very important for UGV in the high-speed driving process.

B. Our work

Our research focuses on the UGV in high-speed driving process in rough terrain. Our method is divided into two steps, the first is single frame obstacle detection and the second is probability map reconstruction based on single frame detection

978-1-7281-9888-0/20/$31.00 ©2020 IEEE

result. Our single frame detection method for obstacles is similar to paper [10] and [11], which 3D point clouds are projected to the 2D overhead view for data processing and uses the 2D grip map to represent local perception map. Gradient statistics in the height direction is an important detection step. Different from paper [11], sliding window and neighborhood relationship is applied to determine the occupancy of obstacle in each position of the grid map rather than just the individual projection grid position of point clouds. Due to the irregular obstacles in rough terrain, in the case of road turbulence, there is uncertainty in obstacle detection, it is helpful to improve the accuracy of obstacle detection combined with historical detection results. Here we use Bayesian probability model to complete fusion of multi-frames detection results. Paper [12] represents a map reconstruction method based on 2D-Lidar. We introduce this method to the application of 3D-Lidar based obstacle detection, and combined with the single frame obstacle detection result, the fusion result can be used for autonomous navigation in real time.

Considering the road turbulence and the unstructured characteristics of obstacle in rough terrain, according to the single frame detection results, by using the first-order Markov model hypothesis, Bayesian probability model based map reconstruction technology is used to guarantee the stability of obstacle detection in unstructured scenes. This method can also improve the robustness of environmental perception. At the same time, map reconstruction can be used to retain the obstacle detection result in the blind area in order to improve the safety of autonomous navigation especially in high-speed.

The paper is organized as follows: section II is system overview, section III is the single frame obstacle detection, section IV is probability map reconstruction, section V is experiments and the last is the conclusion.

II. System Overview

In terms of hardware, Pandar40p is chosen as our 3D-Lidar sensor and the proposed algorithm can also be used with other 3D-Lidar sensors such as Velodyne HDL-64 and RS-Lidar-32, as our algorithm is independent of sensor model. Location data of integrated navigation by GPS and inertial measurement unit (IMU) is necessary to provide the position and attitude information in the process of probability map reconstruction during vehicle movement.

The data processing of the proposed method is divided into two step: single frame obstacle detection and probability map reconstruction as shown in Fig. 1. First, point clouds from Lidar are projected onto the 2D grid map of the overhead view to reduce the time complexity. Then gradient information is used to detect obstacles by statistics. The point clouds are labeled with obstacle ones and obstacle-free ones for the next step of map reconstruction according to the detection result in the former step. We use down-sampling for labeled point clouds to remove redundant data and by coordinate transformation, point clouds are transformed to a unified map coordinate frame associated with historical detection data. After that, Bayesian probability model are used for map updating. The final output of obstacles consists of the single frame detection result and map reconstruction result.

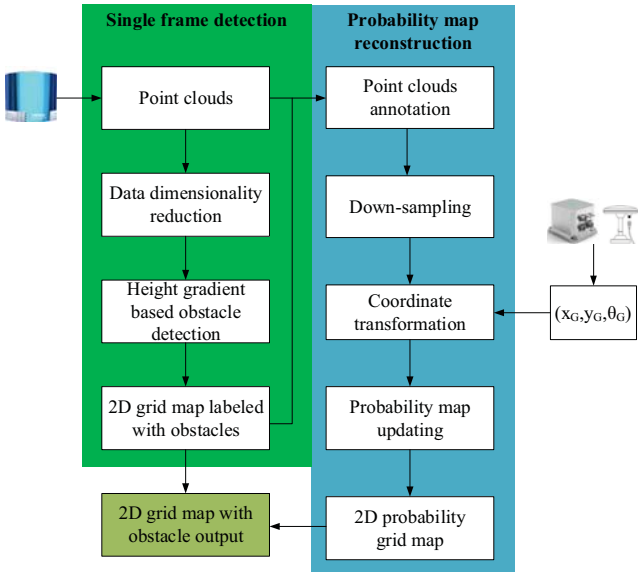


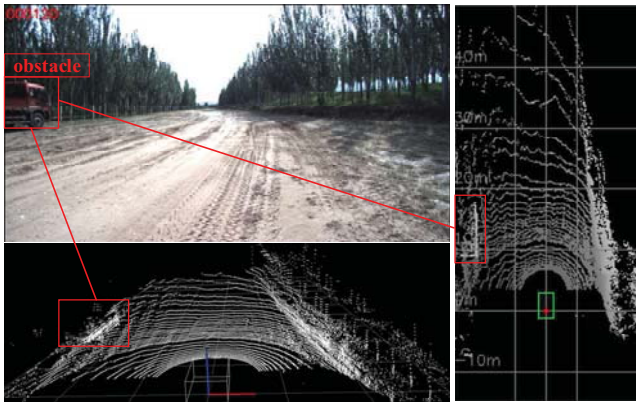
Fig. 1. The flow chart of the proposed algorithm

III. Single Frame Obstacle Detection

Single frame obstacle detection is the precondition of map reconstruction. In this paper, statistics of height gradient information in the 2D grid map is used to extract obstacles. To reduce the noise of point clouds in rough terrain and reduce data amount, some preprocess work is done first.

A. Data dimensionality reduction

Point clouds are discretely distributed in 3D space. Obstacles in unstructured terrain stand on the ground and in the plane space, according to the height information, they can be grouped into one class. According to these, in order to reduce the time complexity of data processing, point clouds are projected to the overhead view defined as 2D grid map. The size of each grid cell is $0.2m^2$/pixel.



(a) (b)

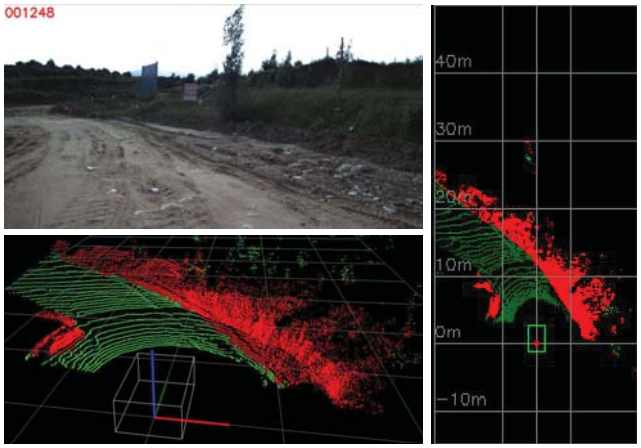
Fig. 2. Demonstration of the projected grid map. (a) Point cloud in 3D space, (b) The projected 2D grid map. The size of the map is 30m *65m with 50m ahead of the vehicle.

As the grid cell may contain projection data of multiple point clouds, point clouds projected to the same grid cell are represented in vector form as $Z_{x,y}=\{z^1_{x,y}, z^2_{x,y}, z^3_{x,y}, \ldots\}$ according to the height information. This dealing can

effectively keep the height information of each point cloud and improve the correlation of adjacent point clouds along the height direction. Fig. 2 shows a demonstration of the projected grid map where the grey value reflects the maximum height of each grid cell. The closer the color is to white, the higher the obstacle's height is. In Fig. 2, the obstacle's color of the labeled vehicle is obviously whiter than the ground's.

B. Height gradient based obstacle detection

In rough terrain, due to the fluctuation of the road, depending on the height information to distinguish obstacles will cause many false detections of obstacles. Fig. 3(a) shows a typical bumpy road environment. Compared with the absolute height, the height gradient can better describe the vertical distribution characteristics of obstacles and has high robustness to rough terrain.



(a) (b)

Fig. 3. Height gradient based obstacle detection. (a) Point cloud in 3D space, (b) The projected 2D grid map. The red area represents obstacle and the green area represents obstacle-free.

First, the lower limit of each grid cell is recorded according to $Z_{x,y}$ and defined as $Z^{min}_{x,y}$. In consideration of the discrete distribution of point clouds, to reduce the missed detection of unstructured obstacles like crushed stone in rough terrain, a sliding window based on *n*-neighborhoods is adopted to determine the reference height of the ground plane defined as $G^{ref}_{x,y}$ shown in (1).

$$G^{ref}_{x,y} = \min\left\{ Z^{\min}_{xx,yy} \mid (xx, yy) \in U^{n}_{x,y} \right\} \tag{1}$$

Where $U^{n}_{x,y}$ represents the *n*-neighborhoods of (x,y). Fig. 4 shows a typical 8-neighborhoods relationship. The gradient of each grid cell defined as $\Delta_{x,y}$ is expressed as the maximum height difference between each point cloud projected to the grid cell and $G^{ref}_{x,y}$ shown in (2).

$$\Delta_{x,y} = \max\left\{ \left\| z^{i}_{x,y} - G^{ref}_{x,y} \right\| \middle| z^{i}_{x,y} \in Z_{x,y} \right\} \tag{2}$$

The gradient threshold value of obstacles is defined as T_Z^{min} and the grid cell with $\Delta_{x,y} > T_Z^{min}$ is regarded as obstacle occupation, otherwise free.

In some cases, such as branches of trees extending to roads, this situation will cause $\Delta_{x,y}$ is largely greater than T_Z^{min} but the road is safe to pass. To solve the problem, the upper gradient limit T_Z^{max} is set according to the actual height of the UGV. Then the solution of $\Delta_{x,y}$ is modified by formula (3).

$$\Delta_{x,y} = \max\left\{ \left\| z^{i}_{x,y} - G^{ref}_{x,y} \right\| \middle| z^{i}_{x,y} \in Z_{x,y}, \left\| z^{i}_{x,y} - G^{ref}_{x,y} \right\| < T_Z^{\max} \right\} \tag{3}$$

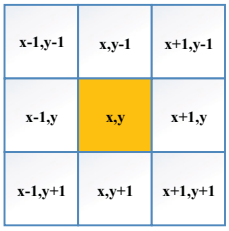


Fig. 4. 8-neighborhoods.

In addition, there is often fugitive dust when the UGV drives in rough terrain. On one hand, we find that the echo reflectivity value of fugitive dust scanned by Lidar is much lower than others. A reflectivity threshold T_R^{min} is set to filter out these noise point clouds. On the other hand, for large particle dust in the air, just depending on the judgement of echo reflectivity, these noise point cloud may not be able to be filtered out clean. Considering that there is usually very rare large particle dust in the air, so by counting the number of point clouds whose gradient meets (4), the number of obstacle point cloud defined as $C_{x,y}$ in each grid cell labeled obstacle occupancy must be greater than a threshold value T_C^{min}. $C_{x,y}$ is calculated by (5).

$$T_Z^{\min} < \left\| z^{i}_{x,y} - G^{ref}_{x,y} \right\| < T_Z^{\max} \tag{4}$$

$$C_{x,y} = \sum_{z^{i}_{x,y} \in Z_{x,y}} \left(T_Z^{\min} < \left\| z^{i}_{x,y} - G^{ref}_{x,y} \right\| < T_Z^{\max} \,?1:0 \right) \tag{5}$$

In the single frame detection processing, the obstacle occupancy of each grid cell $S_{x,y}$ is determined according to (6). Fig. 3(b) shows the single frame obstacle result, where the red area represents obstacles and the green area represents obstacle-free.

$$S_{x,y} = \begin{cases} OBSTACLE, C_{x,y} \geq T_C^{\min} \\ FREE, C_{x,y} < T_C^{\min} \end{cases} \tag{6}$$

When the UGV drives in the case of terrain fluctuation, detection result of the same obstacle at different times are unstable. To improve the stability of detection results, probability statistics for obstacle detection of multi-frames are necessary.

IV. Probability Map Reconstruction

Single frame detection can identify most of obvious obstacles in the sensing range of the Lidar, but due to the bumpy road, the detection results are usually unstable. More seriously, obstacles in the blind area cannot be effectively detected. In order to retain the detection result in the blind area and improve the confidence level of obstacle detection, the probability map reconstruction is established. The map is

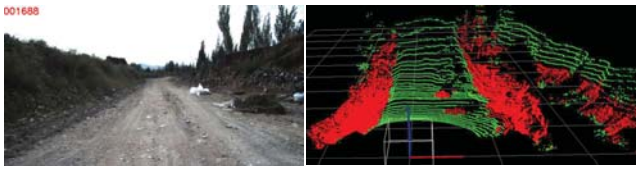
updated by Bayesian probability model according to single frame detection result.

A. Point cloud annotation

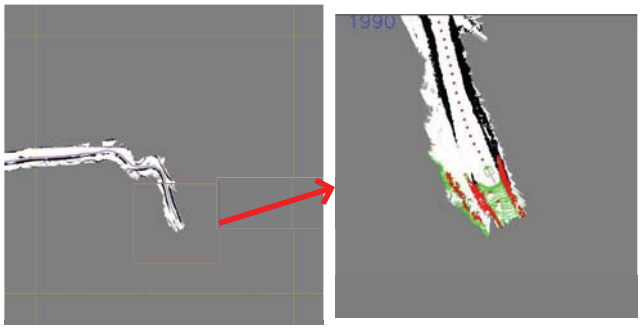
The number of point clouds projected to each cell in the grid map reflects the confidence level of the occupancy of obstacle or obstacle-free. The point clouds are annotated obstacle point cloud P_{obs} and free area point cloud P_{free} according to results of the binary grid map obtained from the single frame data processing as shown in Fig. 3(a). And these labeled point clouds are used as observation data to update the probability map.

B. Down-sampling

Point clouds become sparse as distance increases, and map reconstruction in the near blind area is more important for driving safety. For the above reasons, in our experiment, only point clouds with 20 meters in front of the vehicle are chosen as the region of interest, which can also reduce the amount of data to be processed. In addition, spatial down sampling is used to further reduce the amount of point clouds. Only one point cloud is reserved for each voxel. As point clouds of obstacle are more important than those of obstacle-free. Here the voxel resolution of P_{obs} is set to 0.1m^3 and the voxel resolution of P_{free} is set to 0.2m^3.



(a) Image. (b) Point clouds.



(c) Probability map with size of 500*500m^2. (d) Partial magnification of the map with size of 50*50m^2.

Fig. 5. Probability map reconstruction.

C. Coordinate transformation

In order to reduce the frequent switching of the reconstructed map, we maintain a larger local map. A 2D local probability map whose size is 500*500m^2 is maintained in real time as shown in Fig. 5(c). And its resolution is also 0.2m^2/pixel. The initial location of the UGV is defined as the center of the map M_O. In the process of the UGV's real-time movement, probability map updating is related with coordinate transformation among sensor coordinate frame defined as F_{sensor}, map coordinate frame defined as F_{map} and geodetic coordinate frame defined as F_G. The global coordinate position of the UGV is consistent with INS and GPS. And the position relationship between Lidar and INS is defined as $(\Delta X, \Delta Y)$.

Point clouds in F_{sensor} needs to transformed to the frame of F_{map}. The displacement between the UGV's current position (x_G, y_G, θ_G) and the origin of the probability map $(x_O, y_O, 0)$ is defined as $(x^{\Delta}_G, y^{\Delta}_G, \theta^{\Delta}_G)$. It is expressed by (7).

$$\begin{bmatrix} x_G^{\Delta} \\ y_G^{\Delta} \\ \theta_G^{\Delta} \end{bmatrix} = \begin{bmatrix} x_G \\ y_G \\ \theta_G \end{bmatrix} - \begin{bmatrix} x_O \\ y_O \\ 0 \end{bmatrix} \tag{7}$$

The Bayesian probability model combined with the Bresenham line algorithm is used for map updating. We take Lidar sensor as the origin point and each point cloud as the end point to update the map along the straight line segment direction. Define the location of Lidar sensor in the frame of map as $(x^{map}_{sensor}, y^{map}_{sensor})$ and point cloud in the frame of map as $(x^{map}_{p}, y^{map}_{p})$. The transformation relationship of them among F_{sensor} , F_{map} and F_G are shown in (8) and (9).

$$\begin{bmatrix} x_{sensor}^{map} \\ y_{sensor}^{map} \end{bmatrix} = \begin{bmatrix} x_G^{\Delta} \\ y_G^{\Delta} \end{bmatrix} + \begin{bmatrix} \sin\theta_G & \cos\theta_G \\ -\cos\theta_G & \sin\theta \end{bmatrix} \cdot \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} x_p^{map} \\ y_p^{map} \end{bmatrix} = \begin{bmatrix} x_G^{\Delta} \\ y_G^{\Delta} \end{bmatrix} + \begin{bmatrix} \sin\theta_G & \cos\theta_G \\ -\cos\theta_G & \sin\theta \end{bmatrix} \cdot \begin{bmatrix} x_p^{sensor} + \Delta X \\ y_p^{sensor} + \Delta Y \end{bmatrix} \tag{9}$$

D. Probability map updating

Each grid cell in the probability map is defined as the probability of obstacle occupancy between 0 and 1, where 1 represents the maximum probability of obstacle occupation and 0 represents obstacle-free.

For each point cloud, the straight line area from sensor to the scanned location is considered as obstacle-free. Based on the assumption, the Bresenham line algorithm is used to update the probability map along the straight line segment. Meanwhile for obstacle point cloud, *OBS_PROB* represents a high probability of obstacle occupancy during the observation by sensor and *FREE_PROB* represents a low probability of obstacle occupancy.

Based on the assumption of the first-order Markov model, the state of the probability map at time t is only related to the state at time $t-1$, so map updating only involves the current observation data at time t and the previous map state at $t-1$.

Define $p(m_t)$ as the probability of obstacle occupation in the probability map at time t, and z_t represents the observation data from the current obstacle detection at time t. Since the probability map's updating is independent of the movement of UGV, the influence of vehicle motion is ignored in the probability model. According to Bayesian formula, the update result of the probability map under the latest obstacle detection result at z_t can be described by (10):

$$p(m_t \mid z_{1:t}) = \frac{p(m_{t-1} \mid z_{1:t-1}) \cdot p(z_t \mid m_{t-1})}{p(z_t \mid z_{1:t-1})} \tag{10}$$

Combined with (11), Formula (10) can be rewritten as shown in (12).

$$p(z_t \mid m_{t-1}) = \frac{p(z_t) \cdot p(m_{t-1} \mid z_t)}{p(m_{t-1})} \tag{11}$$

$$p(m_t \mid z_{1:t}) = \frac{p(m_{t-1} \mid z_{1:t-1}) \cdot p(z_t) \cdot p(m_{t-1} \mid z_t)}{p(z_t \mid z_{1:t-1}) \cdot p(m_{t-1})} \tag{12}$$

Meanwhile we can obtain the probability of obstacle-free as shown in (13).

$$p(\overline{m_t} \mid z_{1:t}) = \frac{p(\overline{m_{t-1}} \mid z_{1:t-1}) \cdot p(z_t) \cdot p(\overline{m_{t-1}} \mid z_t)}{p(z_t \mid z_{1:t-1}) \cdot p(\overline{m_{t-1}})} \tag{13}$$

According to $p(\overline{m_t} \mid z_{1:t}) = 1 - p(m_t \mid z_{1:t})$, combined with (12) and (13), we can obtain the relationship among $p(m_t|z_{1:t})$, $p(m_{t-1}|z_{1:t-1})$ and $p(m_{t-1}|z_t)$ shown in (14):

$$\frac{p(m_t \mid z_{1:t})}{1 - p(m_t \mid z_{1:t})} = \frac{p(m_{t-1} \mid z_{1:t-1}) \cdot p(m_{t-1} \mid z_t) \cdot (1 - p(m_{t-1}))}{(1 - p(m_{t-1} \mid z_{1:t-1})) \cdot (1 - p(m_{t-1} \mid z_t)) \cdot p(m_{t-1})} \tag{14}$$

In (14), $p(m)$ means the probability of undetected grid cell, so $p(m)$=0.5. Thus, Formula (14) can be simplified as (15):

$$\frac{p(m_t \mid z_{1:t})}{1 - p(m_t \mid z_{1:t})} = \frac{p(m_{t-1} \mid z_{1:t-1}) \cdot p(m_{t-1} \mid z_t)}{(1 - p(m_{t-1} \mid z_{1:t-1})) \cdot (1 - p(m_{t-1} \mid z_t))} \tag{15}$$

Further, define S_t as shown in (16),

$$S_t = \frac{p(m_{t-1} \mid z_{1:t-1}) \cdot p(m_{t-1} \mid z_t)}{(1 - p(m_{t-1} \mid z_{1:t-1})) \cdot (1 - p(m_{t-1} \mid z_t))} \tag{16}$$

Then the result of map updating can be obtained by (17).

$$p(m_t \mid z_{1:t}) = \frac{S_t}{1 + S_t} \tag{17}$$

Here to reduce the complexity of probability calculation, and considering observation error, for each grid cell with detected result in real time, we define $p(m_{t-1}|z_t=obs)$= *OBS_PROB*, conversely grid cell with detected obstacle-free is $p(m_{t-1}|z_t=free)$= *FREE_PROB=1- OBS_PROB*. For area in the line segment between the sensor and each point cloud, the probability is also regarded as obstacle-free.

Based on the above derivation, the Bresenham line algorithm is used to update the line segment area from the sensor position (x^{map}_{sensor},y^{map}_{sensor}) to each point cloud position (x^{map}_{p},y^{map}_{p}). The specific pseudo code is shown in Fig. 6.

```
01. (x,y) ←(x^map_sensor,y^map_sensor)
02. (Δx, Δy)←(|x^map_p - x^map_sensor|,| y^map_p - y^map_sensor |)
03. (sx,sy) ← (x^map_p - x^map_sensor ≥0?1:0, y^map_p - y^map_sensor ≥0?1:0)
04. If Δy ≥Δx
05. Then swap(Δx, Δy)
06.         swapFlag←TRUE
07. Else swapFlag ←FALSE
08. p ←2*Δy-Δx
09. For i ←1:1: Δx
10.     If i==Δx
11.     Then
12.         If z_t(x,y)==obs
13.         Then UpdateMap(x,y,OBS_PROB)
14.         Else UpdateMap(x,y,FREE_PROB)
15.     Else    UpdateMap(x,y,FREE_PROB)
16.     If p ≥0
17.     Then
18.         If swapFlag==TRUE
19.         Then x ←x+sx
20.         Else y ←y+sy
21.         p ←p-2*Δx
22.     If swapFlag==TRUE
23.     Then y ←y+sy
24.     Else x ←x+sx
25.     p ←p+2*Δy

26. UpdateMap(x,y,PROB)
27.     S(x,y) ←p^prior_x,y /(1- p^prior_x,y)*PROB/(1-PROB)
28.     P^update_x,y ←S_x,y /(1+S_x,y)
```

Fig. 6. The pseudo code of probability map updating.

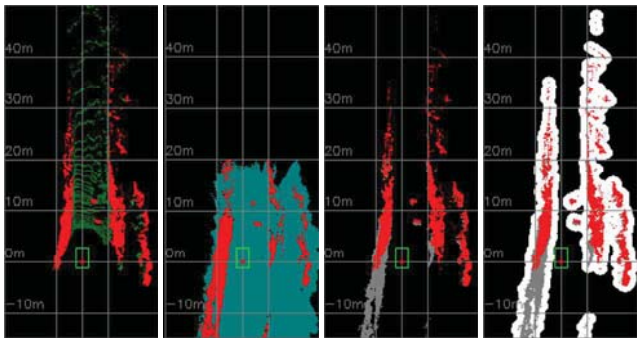
Line 16 to Line 25 shows the calculation method of the next grid cell on the line by the Bresenham line algorithm. The map updating from Line 27 to Line 28 is based on (16) and (17).

Fig. 5(d) shows the map updating process, where the red area represents the projection location of current point clouds of obstacles, the green area represents the projection location of current point clouds of obstacle-free. The grey value of map reconstruction area reflects the probability of obstacle occupancy. The color close to black represents a high probability with obstacles, the color close to white represents a high probability with obstacle-free and the color of light grey with the largest area represents unknown area.

According to statistics of a large number of test result, for the probability map, the probability of grid cell with $p(m_t|z_{1:t})$>0.7 is taken as the obstacle occupancy grid and the other area including unknown area is taken as obstacle-free. As shown in Fig. 7(b), the cyan area represents obstacle-free, the red area represents obstacles, and the black area represents

undetected area by sensor. The map is generated according to Fig. 5(c).

Probability map reconstruction will still retain the previous detection result even when obstacles enter the blind area of sensor as previous obstacle detection result are included in the map updating processing. Obstacles retention in the blind area is essential for the safety of autonomous navigation.



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

Fig. 7. The output of obstacles. (a) Single frame obstacle detection. (b) Result of the probability map. (c) Fusion result with (a) and (b). (d) Fusion result with Obstacles expansion.

The final output of obstacles consists of the result in the probability map and the single frame detection as shown in Fig. 7(c). The red area represents the current frame obstacle detection result of Fig. 7(a) and the grey area represents the reconstruction result based on the probability map in Fig. 7(b). In Fig. 7(d), the white area around obstacles represents the expansion according to the vehicle's size, which is common used in the local navigation map of the UGV.

V. Experiments

We have done a lot of experiments to verify the proposed algorithm in rough terrain. The Lidar sensor we choose is pandar40p. The sensor is mounted on the roof of the vehicle. The basic performance of the computing platform includes: Intel I7 6700HQ@2.6GHz, RAM 8.00GB, NVIDIA GeForce GTX 960M. The software runs on Ubuntu 16.04.

The experiment environment involves pavement road, dirt road with scattered obstacles such as wooden piles, cone barrels and other obstacles in nature. Several important parameters mentioned above are set as follows in our experiment: T_Z^{min} =0.3, T_Z^{max} =2.0, T_C^{min} =5, *OBS_PROB*=0.85, *FREE_PROB* =0.45.

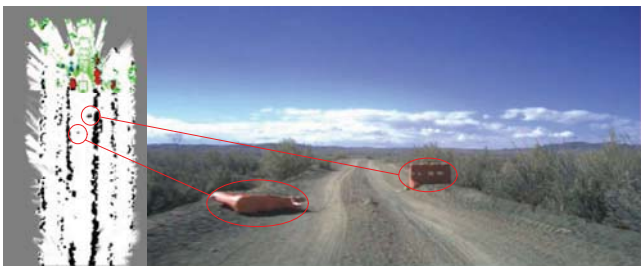


Fig. 8. Local map reconstruction.

To verify the validity of the proposed method for obstacle detection, two obstacles are randomly placed on the ground and a planned route is designed for the UGV to drive through this area autonomously. In Fig. 8, after the two obstacles entered the blind area behind the vehicle, the result of obstacle reconstruction can be clearly retained shown in Fig. 8. And the UGV can safely pass through the area without obstacle collision.

The proposed method can also be used for unknown scene mapping of any size not just within 500*500m^2. Through map stitching, local map with size of 500*500m^2 can be stored and reused in the form of map blocks. Fig. 9 shows a large-scale map reconstruction result of a field test site. The topological connection relationship of these roads can described clearly in the rebuilt map.

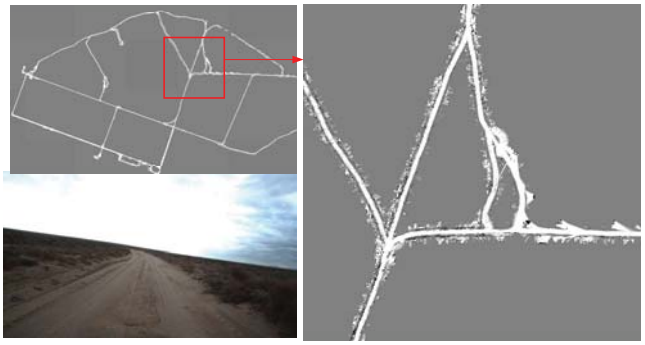


Fig. 9. Large-scale map reconstruction.

The experiment results show that the proposed method has a good detection effect on obstacles in rough terrain. The map reconstruction ensures the retention of obstacles in the blind area. The time complexity of the proposed method is about 90ms through a series of data redundancy processing, and it meets the real-time requirement for autonomous navigation.

VI. Conclusion

This paper focuses on obstacle detection in rough terrain. First, gradient statistics based single frame detection can effectively extract obstacles in the bumpy road and reduce false detection. Then by using Bayesian probability model based map reconstruction, it can retain obstacle information in the blind area in order to guarantee the safety of the UGV's autonomous navigation especially in high speed. The proposed method can be applied to the environment awareness of the UGV's autonomous navigation. And the map reconstruction technology can also be used for rapid map building in unknown scenes.

However, the current research mainly studies on positive obstacle detection while properties of obstacles are not distinguished. In the further research, we will work on the classification of detail obstacles such as vehicles, pedestrians, trees and others. Each type of obstacles will be labeled with the unique attribute tag. At the same time, dynamic obstacles will be screened during the process of map reconstruction to ensure that only static obstacles are remained in the reconstruction map. This work is of great significance for the construction of unknown scene map and enhance scene understanding.

REFERENCES

[1] Zych N, Silver D, Stager D, et al. Achieving integrated convoys: cargo unmanned ground vehicle development and experimentation[C]//SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2013: 87410Y-87410Y-14.

[2] D. Droeschel, M. Schwarz, S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner", Robotics and Autonomous Systems, vol. 88, pp. 104-115, 2017.

[3] Wang X , Wang J , Zhang Y , et al. 3D LIDAR-Based Intersection Recognition and Road Boundary Detection Method for Unmanned Ground Vehicle[J]. 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC).

[4] Asvadi A, Premebida C, Peixoto P, et al. 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes[J]. Robotics and Autonomous Systems, 2016, 83: 299-311.

[5] Zhang J , Singh S . Low-drift and real-time lidar odometry and mapping[J]. Autonomous Robots, 2017, 41(2):401-416.

[6] Wu B, Wan A, Yue X, et al. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 1887-1893.

[7] Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 770-779.

[8] Chen, Xiong, and Zhidong Deng. "Detection of Road Obstacles Using 3D Lidar Data via Road Plane Fitting." Proceedings of the 2015 Chinese Intelligent Automation Conference. Springer, Berlin, Heidelberg, 2015.

[9] Wu B, Wan A, Yue X, et al. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 1887-1893.

[10] Suger B, Steder B, Burgard W. Terrain-adaptive obstacle detection[C]//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 3608-3613.

[11] Ning L I , Jiang-Hua G , Wei L . Environment Perception Research Based on 3D-lidar in the Unstructured Road[J]. Vehicle & Power Technology, 2017.

[12] Hu Y , Gong J , Jiang Y , et al. Hybrid Map-Based Navigation Method for Unmanned Ground Vehicle in Urban Scenario[J]. Remote Sensing, 2013, 5(8):3662-3680.