

# Towards an obstacle detection system for robot obstacle negotiation

Han Wang, Quan Zhang, Zhenquan Fan, Gongcheng Wang, Pengchao Ding and Weidong Wang  
Harbin Institute of Technology, Harbin, China

## Abstract

**Purpose** – To solve the obstacle detection problem in robot autonomous obstacle negotiation, this paper aims to propose an obstacle detection system based on elevation maps for three types of obstacles: positive obstacles, negative obstacles and trench obstacles.

**Design/methodology/approach** – The system framework includes mapping, ground segmentation, obstacle clustering and obstacle recognition. The positive obstacle detection is realized by calculating its minimum rectangle bounding boxes, which includes convex hull calculation, minimum area rectangle calculation and bounding box generation. The detection of negative obstacles and trench obstacles is implemented on the basis of information absence in the map, including obstacles discovery method and type confirmation method.

**Findings** – The obstacle detection system has been thoroughly tested in various environments. In the outdoor experiment, with an average speed of 22.2 ms, the system successfully detected obstacles with a 95% success rate, indicating the effectiveness of the detection algorithm. Moreover, the system's error range for obstacle detection falls between 4% and 6.6%, meeting the necessary requirements for obstacle negotiation in the next stage.

**Originality/value** – This paper studies how to solve the obstacle detection problem when the robot obstacle negotiation.

**Keywords** Obstacle detection, Elevation map, Obstacle negotiation

**Paper type** Research paper

## 1. Introduction

Mobile robots equipped for obstacle negotiation are commonly used in applications such as rescue and disaster relief (Wang *et al.*, 2023; Bai *et al.*, 2023a, 2023b), scientific exploration (Zhu *et al.*, 2021) and military operations due to their robust environmental adaptability (Bai *et al.*, 2023a, 2023b). Obstacle negotiation is the premise for robots to overcome obstacles (Zhu *et al.*, 2021), and the methods for mobile robots to achieve autonomous obstacle negotiation can be divided into two categories: environment-based motion planning obstacle negotiation method (Li *et al.*, 2016) and information-based online control obstacle negotiation method (Vincent and Sun, 2012; Sokolov *et al.*, 2017). Either method is inseparable from the perception of the surroundings (Xie *et al.*, 2018; Gao *et al.*, 2021a, 2021b), and the environment-based motion planning obstacle negotiation method has higher requirements for obstacle recognition (Gao *et al.*, 2022, 2021a, 2021b). The result of obstacle detection largely determines the success or failure of autonomous obstacle negotiation.

Therefore, obstacle detection is crucial for autonomous obstacle negotiation.

Owing to the difference in the method of implementation, obstacle detection can be divided into sensor-based and map-based methods. Among the sensor-based methods, the vision-based method (Shi *et al.*, 2022; Mancini *et al.*, 2018) was

developed earlier, but this method has a limited ability to detect the specific size of obstacles. Naujoks *et al.* (Naujoks and Wuensche, 2018) calculated the three-dimensional direction correction frame of an obstacle using point cloud data of a three-dimensional lidar, which can obtain the spatial pose and size of obstacles. Based on the geometric characteristics of obstacles, Larson and Trivedi (2011) and Shang *et al.* (2014) achieved the identification of negative obstacles through the relationship between the data of 3D lidar scan lines. The sensor-based methods perform obstacle detection based on the sensor's limited frame data, which is likely to cause problems such as incomplete recognition results. These drawbacks can be avoided by map-based methods, of which elevation maps (Florin *et al.*, 2007; Drulea *et al.*, 2016) are used more. Dornhege and Kleiner (2007) established a behavior map for obstacle crossing based on the elevation map, which stores a set of skill descriptions corresponding to the robot in the grid according to the elevation information. Wermelinger *et al.* (2016) built a traversability map for quadruped robot navigation and obstacle crossing based on the elevation map, in which the traversability is determined by the weighting of factors such as slope, roughness and steps of terrain. In the existing methods, there are few methods that can simultaneously detect positive obstacles and negative obstacles and can calculate specific information such as sizes.

The current issue and full text archive of this journal is available on Emerald Insight at: <https://www.emerald.com/insight/0143-991X.htm>



Industrial Robot: the international journal of robotics research and application  
51/2 (2024) 236–245  
© Emerald Publishing Limited [ISSN 0143-991X]  
[DOI 10.1108/IR-09-2023-0210]

This work is supported in part by the National Natural Science Foundation of China under Grant 62273123.

Received 8 September 2023

Revised 11 October 2023

27 October 2023

6 November 2023

22 November 2023

Accepted 23 November 2023

We propose an obstacle detection system based on elevation maps that covers the entire process, from environmental mapping to obstacle recognition and condition analysis of obstacle crossing. The system can detect obstacle sizes with high efficiency and accuracy in a range of scenarios. The research mainly focuses on three types of obstacles: positive obstacles, negative obstacles and trench obstacles. Positive obstacles refer to obstacles above the ground where the robot is currently located. Negative obstacles refer to areas below the ground, such as obvious sinking terrain. Trench obstacles refer to terrain with a certain span and depth, and the front and rear edges are approximately parallel. The system will mainly solve the problem of detecting the above three types of obstacles. For positive obstacles, an algorithm based on the smallest rectangular bounding box is proposed. For negative obstacles and trench obstacles, the algorithm is based on the information absence in the elevation map to initially locate them, and then through further detection to determine their final type. Generally speaking, this work provides the robot with the ability to determine whether it has the conditions to overcome obstacles, thereby supporting the robot to perform high-level command tasks such as obstacle navigation in complex environments. It has high application value in the field of robotics.

This paper is organized as follows. After giving a brief review of the related work in Section 2, we will introduce the principles and technical details of the system in Section 3. Section 4 then presents related evaluation experiments. Finally, conclusions and prospects will be drawn in Section 5.

## 2. Related work

### 2.1 Elevation mapping

Compared with two-dimensional map, 2.5D elevation map can express richer ground elevation information. At the same time, the calculation and storage are more efficient than 3D map. The early elevation mapping was mainly realized by sparse two-dimensional lidar data (Ye and Borenstein, 2003; Belter and Skrzypczyński, 2011), while using a filter to fill the elevation map to deal with the problem of invalid and missing original data (Ye and Borenstein, 2004). Recently, approaches to elevation mapping in rough terrain have tended to focus either on probabilistic estimating (Belter et al., 2012; Fankhauser et al., 2018) and considering the measurement noise of the sensor and the uncertainty of robot localization (Fankhauser et al., 2014), or on efficient update of elevation map (Kleiner and Dornhege, 2007). These approaches provide a strong guarantee for the map-based obstacle detection methods.

### 2.2 Ground segmenting

To calculate the specific location and size of the obstacle, the ground must be segmented. A grid-based approach was introduced, which divides the grid cells into ground and nonground according to the maximum absolute difference between the height of points in the grid. Similarly, (Chu et al., 2017) provides a fast and effective method for ground segmentation. It determines the threshold point depending on three features: gradient, lost threshold points and abnormalities in the distance between the sensor and a particular threshold point. Then, it estimates ground points and nonground points

based on threshold points. On the other hand, Moosmann et al. (2009) create an undirected graph and compare local changes in the plane normal to characterize the change in slope.

*Obstacle clustering:* The approaches used for obstacle clustering are k-medoid algorithm (Hou, 1999), AK-means clustering algorithm (Xu et al., 2012), density-based approach (Zaiane and Lee, 2002) and elevation-reference connected component labeling algorithm (Tian et al., 2020), etc. The idea of obstacle clustering we adopted is very close to the Euclidean clustering algorithm. Euclidean clustering algorithm is a method for neighborhood clustering of points in three-dimensional space.

### 2.3 Obstacle recognition

In the field of obstacle recognition, there are common positive obstacle recognition and negative obstacle recognition. Ortigosa et al. (2011) recognizes positive obstacles by detecting whether the column direction depth of the image column matches the linear model. Labayrade et al. (2002) propose the concept of V parallax to identify positive obstacles. In the V parallax space, the ideal road plane and vertical obstacles are shown as an inclined straight line and vertical line, respectively. Hu and Uchimura (2005) further introduce the concept of U-V parallax to adapt to a more general environment. For negative obstacles, methods based on synthetic aperture radar interference (InSAR) and based on multisensor fusion are more common (Wang et al., 2016; Yuan et al., 2015; Forouher et al., 2016). Our obstacle detection system takes multiple scenarios and multiple obstacle types into account, and uses methods based on the minimum rectangle bounding box and based on the characteristics of data absence to identify three types of obstacles.

## 3. Methods

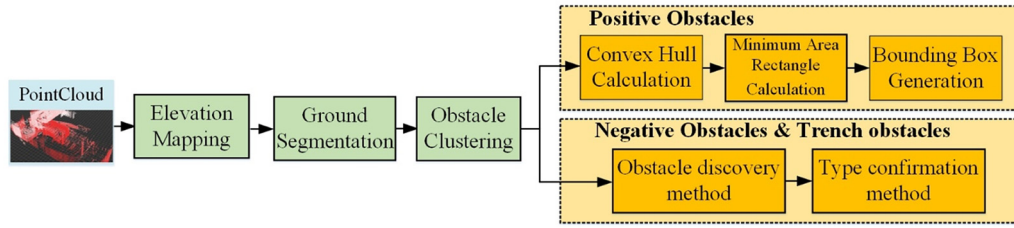
The detection algorithm proposed in this paper, as shown in Figure 1, conducts obstacle recognition research on the basis of elevation maps, including ground segmentation algorithms, obstacle clustering algorithms and specific recognition algorithms for positive obstacles, negative obstacles and trench obstacles.

### 3.1 Elevation mapping and updating

This paper will take the robot's posture and sensor's point cloud data as input, and develop an elevation map algorithm based on Grid Map Library (Fankhauser and Hutter, 2016). In this paper, the setting of the coordinate system for mapping is similar to (Fankhauser et al., 2018), and a local elevation map that follows the movement of the robot is established. This method can avoid the discontinuity of the mapping as much as possible, as shown in Figure 2. In the algorithm, the associated coordinate system is related to the world coordinate system, and the relationship is shown xyz in (1):

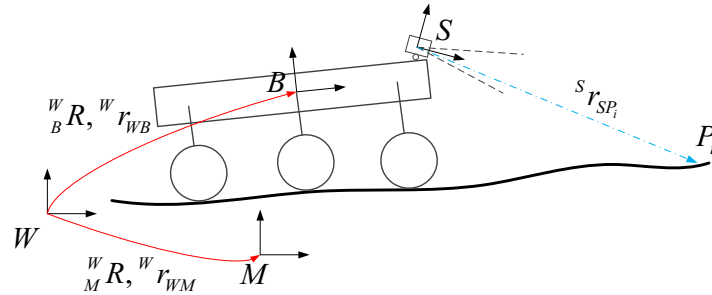
$$\begin{cases} {}^W R(\psi, \theta, \varphi) = I \\ {}^W r_{WM}(x, y) = {}^W r_{WB}(x, y) \\ {}^W r_{WM}(z) = 0 \end{cases} \quad (1)$$

where,  $x, y, z$  and  $\psi, \theta, \varphi$  are, respectively, the translation and rotation angles along each axis among the coordinate systems.

**Figure 1** Working process of obstacle detection system

**Notes:** The positive obstacle detection is realized by calculating its minimum rectangle bounding boxes. The detection of negative obstacles and trench obstacles is implemented on the basis of information absence in the map, including obstacles discovery method and type confirmation method

**Source:** Authors' own work

**Figure 2** Coordinate relationship when elevation mapping

**Notes:**  $W$ ,  $B$  and  $S$  correspond to the fixed world coordinate system, robot coordinate system and sensor coordinate system, respectively.  $M$  is the associated coordinate system of the elevation map

**Source:** Authors' own work

$R$  represents the rotation transformation matrix,  $r$  represents the translation transformation and  $I$  is the unit matrix. The elevation map  $M_{ele}$  constructed in this paper can be expressed by (2):

$$M_{ele} = \left\{ \sum_{i=1}^m c_i | c_i = (x_i, y_i, h_i, \sigma_i^2) \right\} \quad (2)$$

where  $c_i$  is the grid point in the elevation map, whose quantity is  $m$ .  $x_i$  and  $y_i$  is the position of the grid point  $c_i$  in the world coordinate system, corresponding to the index  $i$  in the map,  $h_i$  is the elevation value at this grid point and  $\sigma_i^2$  is the variance of the elevation value.

The general idea of our algorithm is the same as that in Fankhauser et al. (2018), which will consider the measurement noise of the sensor and the localization uncertainty of the robot, and the mapping process is divided into two stages: measurement update and movement update. Different from the literature Fankhauser et al. (2018), we have improved the variance calculation formula of measurement update and movement update formula, corresponding to (4) and (5), respectively:

- **Measurement update.** The relationship between the sensor scan point  $^S P_i$  converted to the elevation value  $p_i$  in the map and the variance formula of the elevation value are shown in (3) and (4), respectively, where  $p = [0 \ 0 \ 1]$  is a mapping vector that extracted the  $z$  value of the point

cloud as the elevation value at that point.  $\Sigma_S$  is the covariance matrix measured by the sensor.  $\Sigma_R$  and  $\Sigma_r$  are the position posture covariance matrix and position covariance matrix, respectively. The solution method of the Jacobian matrix and the update process based on the one-dimensional Kalman filter are the same as in Fankhauser et al. (2018).

$$p_i = P \left( {}^M_S R^S r_{SP_i} + {}^M r_{MS} \right) \quad (3)$$

$$\sigma_{p_i}^2 = \mathcal{J}_S \Sigma_S \mathcal{J}_S^T + \mathcal{J}_R \Sigma_R \mathcal{J}_R^T + \mathcal{J}_r \Sigma_r \mathcal{J}_r^T \quad (4)$$

- **Movement update.** To reduce the amount of calculation and simplify the update process, this paper linearly combines the uncertainty of positioning and the corresponding amount of motion, and only updates the variance of the elevation value at a certain frequency. The elevation value remains unchanged during this process, updated equation is as follows:

$$\begin{cases} h_{k+1} = h_k \\ \sigma_{k+1}^2 = \sigma_k^2 + |x_{k+1} - x_k| \Sigma_{r,k+1} (1, 1) + |y_{k+1} - y_k| \Sigma_{r,k+1} (2, 2) \end{cases} \quad (5)$$

where,  $k$  and  $k + 1$  represent two adjacent update moments,  $|x_{k+1} - x_k|$  and  $|y_{k+1} - y_k|$  respectively, represent the

displacement of the robot in the corresponding direction in the world coordinate system.

### 3.2 Ground segmentation based on plane fitting

Ground segmentation is a necessary link before obstacle recognition, used to eliminate the interference of ground information on obstacle detection. In this paper, the ground segmentation algorithm based on plane fitting proposed in Zermas *et al.* (2017) is used to separate the ground and nonground parts in the elevation map.

Because the operating environment of the robot in this paper is three-dimensional, the ground part is not always on the map. So, we made two improvements to the algorithm: the current height of the robot is used as the reference point for the selection of seed points, so that the ground will change with the movement of the robot, which can ensure that the obstacles above the ground are positive obstacles relative to the robot and those below the ground height are negative obstacles. After segmenting the original ground, for areas where obstacles exist, the theoretical ground height will be calculated based on the final ground model.

### 3.3 Obstacle clustering

We converted the Euclidean-clustering algorithm from three-dimensional to two-dimensional for this paper.

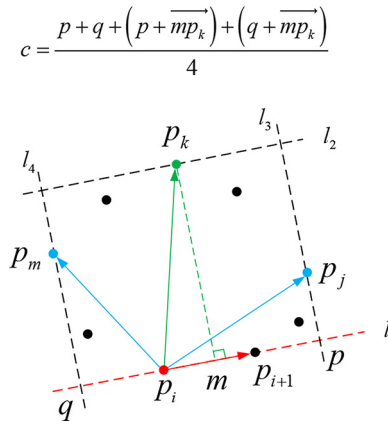
### 3.4 Positive obstacle detection based on minimum bounding box

We realize the recognition of positive obstacles by calculating the three-dimensional bounding box of them. The whole process can be divided into three steps: convex hull, minimum area rectangle and bounding box:

- 1 *Convex hull calculation.* In this paper, the Graham scanning method (Hu and Uchimura, 2005) is used to calculate the convex hull according to the grid coordinates of the elevation map occupied by positive obstacles.
- 2 *Minimum area rectangle calculation.* The minimum area rectangle refers to the smallest circumscribed rectangle formed by convex hull points, which is closest to the actual shape of the obstacle. The general idea of the calculation of the minimum area rectangle is: connecting two adjacent points in the convex hull to form a straight line  $l_1$ , using this line as the bottom edge, searching for the parallel line  $l_2$  that passes other convex hull points and is farthest from  $l_1$  and two perpendicular lines  $l_3$  and  $l_4$  farthest apart from each other. The four straight lines intersect to form a rectangle. Traverse the convex hull and connect two adjacent points in sequence to obtain different bottom edges, and then obtain a rectangle corresponding to it, in which the smallest area is the desired one. After the bottom edge is determined, how to efficiently find the other three edges is the key to the calculation of this part. We use the corner judgment method (Arnon and Gieselmann, 1983) to solve this problem. It can complete the traversal calculation in linear time.

The calculation method of the rectangular area in this paper is different from that in Graham (1972). As shown in Figure 3, the key to the area calculation is to solve the points  $p$ ,  $q$  and  $m$ .

**Figure 3** Rectangular area calculation in positive obstacle detection



**Note:** The black dots in the figure are convex hull points

**Source:** Authors' own work

These three points can be obtained by the nature of vector dot product. Taking point  $p$  as an example, the formula is as follows:

$$p = p_i + \frac{\overrightarrow{p_i p_j} \cdot \overrightarrow{p_i p_{i+1}}}{|\overrightarrow{p_i p_{i+1}}|^2} \overrightarrow{p_i p_{i+1}} \quad (6)$$

According to the meaning of vector dot product, (6) can be written as:

$$p = p_i + \frac{\overrightarrow{p_i p_j} \cdot \overrightarrow{p_i p_{i+1}}}{|\overrightarrow{p_i p_{i+1}}|^2} \overrightarrow{p_i p_{i+1}} \quad (7)$$

The formula for solving the position of the center point of the rectangle is as follows:

$$c = \frac{p + q + (p + \overrightarrow{mp_k}) + (q + \overrightarrow{mp_k})}{4} \quad (8)$$

Taking the straight line  $l_1$  passing through points  $p_i$  and  $p_{i+1}$  as the bottom, the dotted lines intersect to form a rectangle, whose area is the product of the length and width. The length is the length of  $pq$ , and the width is the length of  $p_k m$ , where  $p$ ,  $q$  and  $m$  are the projection points of  $p_j$ ,  $p_m$  and  $p_k$  on line  $l_1$ . (Source: Authors' own work)

#### 3 Bounding box generation

Take the minimum area rectangle as the bottom surface and the maximum height of the obstacle relative to the ground as high, a bounding box can be generated.

### 3.4 Negative obstacle and trench detection

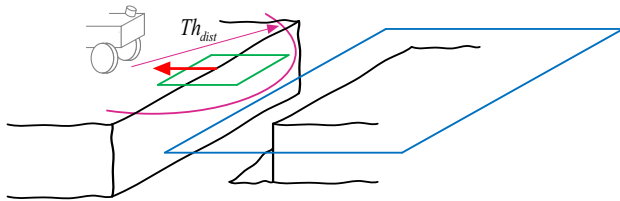
- *Obstacle discovery.* The purpose of this step is to determine whether there are potential negative obstacles or trench obstacles, and to calculate the location of the potential obstacle edge. In this paper, multiple detection units are designated in the map space to detect negative obstacles and trench obstacles. A detection unit contains a detection base point and a detection area associated with the base point.

The following describes the detection process by taking the detection unit closest to the robot shown in Figure 4 as an example. Since the sensor blind area also shows no elevation value in the map, certain conditions need to be made when clustering grid points, which can be described as follows: when clustering starts, the traversal area is only limited to the green rectangle. For grid points belonging to negative obstacles or trench obstacles, the Euclidean distance from the point to the detection base point must be less than a certain distance threshold  $Th_{dist}$ .

When there is a negative obstacle or a trench obstacle in the detection unit, the cluster area will be similar to the purple fan-shaped obstacle area in Figure 5. If the coordinates of points  $v_1$  and  $v_2$  can be calculated, the position  $C$  of the obstacle's edge and the normal  $V$  of the obstacle can be calculated. Based on the shape characteristics of the fan-shaped area, two points can be found by sorting and filtering the points in the fan-shaped area. The process is as follows:

- Convert the two-dimensional coordinates of the grid points in the obstacle area to the robot coordinate system  $R$  and put them into the queue  $P_{unknown}$ .
- Sort  $P_{unknown}$  according to the rule of increasing  $X$  coordinate value. Points with the same  $X$  coordinate value are arranged from small to large according to the  $Y$  coordinate value. The first element of the team is the vertex  $v_1$ , and  $v_1$  is the smaller  $X$  coordinate value of the two vertices.
- Then sort  $P_{unknown}$  according to the rule of increasing  $Y$  coordinate value. Points with the same  $Y$  coordinate value

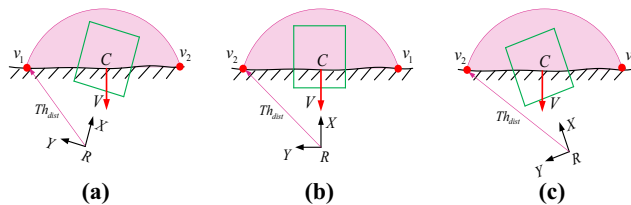
**Figure 4** Closest detection unit to the robot



**Notes:** The detection base point is the center of robot, and the detection area is the green rectangle in front of the robot

**Source:** Authors' own work

**Figure 5** Obstacle edge recognition



**Notes:** The purple fan shape is the potential area of the obstacle.  $R$  is the position of the robot. The point  $v_1$  and  $v_2$  are the two intersections of the area and the edge of the obstacle. (a), (b) and (c) are the left diagonal, positive and right oblique to the obstacle

**Source:** Authors' own work

are arranged from small to large according to the  $X$  coordinate value. If the head element and  $v_1$  are the same point or the Euclidean distance between the two points is small, take the tail element as the vertex  $v_2$ , corresponding to the situation in Figure 5 (b) and Figure 5 (c). Otherwise, take the head element as  $v_2$ , corresponding to the situation in Figure 5 (a).

The angle  $\theta$  between the normal  $V$  and the forward direction of the robot and the position  $C$  are obtained by the following formula:

$$\begin{cases} C = \frac{v_1 + v_2}{2} \\ \theta = -\arctan \frac{v_{2,x} - v_{1,x}}{v_{2,y} - v_{1,y}} + \pi \end{cases} \quad (9)$$

When the width  $|v_1 v_2|$  of the obstacle's edge is greater than the wheelbase of the robot, it is necessary to further confirm the type of obstacle and enter the next link for secondary detection.

- **Type confirmation.** The purpose of this step is to determine whether the obstacle is a negative obstacle or a trench obstacle. The secondary detection area is still rectangular, as shown by the blue rectangular frame in Figure 4. The location of the rectangle is related to point  $C$ .

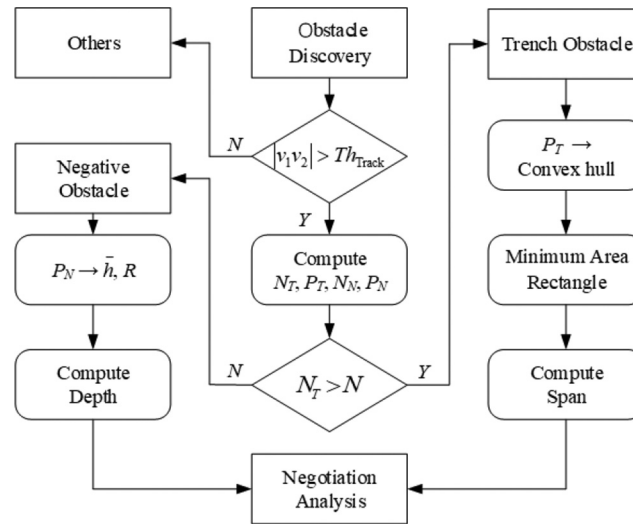
The height of the terrain on both sides of the trench in the trench obstacle tends to be consistent, and the terrain height of the negative obstacle is much lower than the height of the ground where the robot is located. We use this to confirm the type of obstacle. The specific process is as follows:

- Traverse the secondary detection area to find grid points with elevation values, and set the height judgment threshold  $Th$  based on the height of the detection base point. When the height value of the grid points is within the range of  $Th$  above and below the reference. The points are classified as a trench obstacle point set  $P_T$ , otherwise they are classified to the negative obstacle point set  $P_N$ .
- After the traversal, judge the number  $N_T$  of the points in the trench obstacle point set  $P_T$ . If  $N_T$  exceeds the set threshold  $N$ , the place is considered to be a trench obstacle. The convex hull algorithm in  $D$  and the minimum area rectangle algorithm are used to solve the minimum area rectangle of point set  $P_T$ , and the span of the trench can be further calculated.
- If  $N_T$  does not exceed the threshold  $N$ , it is regarded as a negative obstacle and the average height  $\bar{h}$  of the point set  $P_N$  is calculated. The difference between  $\bar{h}$  and the height of the identified base point is taken as the depth of the negative obstacle.

In summary, the detection process of negative obstacles and trench obstacles is shown in Figure 6.

## 4. Experiments

The proposed detection system was tested on the ROS-Gazebo simulation platform, indoor real environment and outdoor real environment. The validation is performed in three scenarios with different types of mobile robot platforms, namely, six-wheeled, tracked and four-wheeled. The resolution of the elevation map is 0.1 m. The feasibility of the

**Figure 6** Negative obstacle and trench obstacle detection process

**Notes:** In the detection process, the potential areas of obstacles are first searched, and then the specific types of obstacles are judged according to their features. Finally, the specific and sizes of negative obstacles and trench obstacles are calculated, respectively

**Source:** Authors' own work

algorithm is demonstrated by testing the success rate of detecting obstacles, and the accuracy of the algorithm is demonstrated by detecting the length or depth of the obstacles. Each scenario was repeated multiple times to ensure the validity of the data and conclusions. The success rate is only used to judge the type of obstacle. For irregular obstacles, the outermost boundary is used as the size standard for judging the accuracy.

#### 4.1 Validated in Gazebo simulation world

The obstacle detection system was tested in simulations of various obstacles. The proposed detection algorithm is carried on a six-wheeled mobile robot. Figure 7 (top) shows five scenarios containing three types of obstacles constructed from various barricades. Figure 7 (middle) shows the effect of elevation mapping. Figure 7 (bottom) shows the detection results of the obstacle detection system.

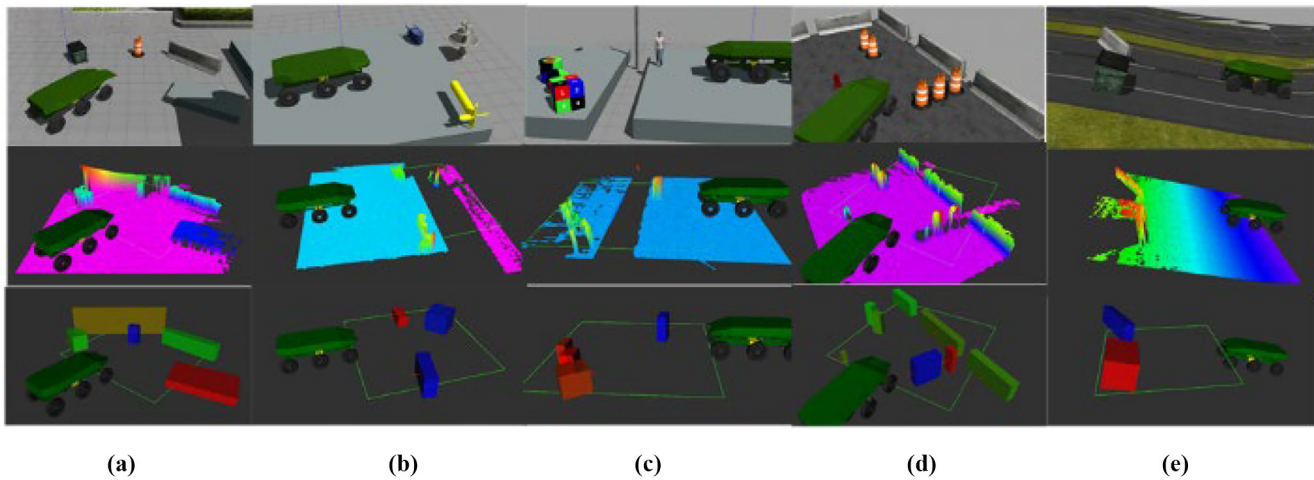
The detection results are shown in Table 1. It can be seen from the figure that the proposed algorithm successfully detects various obstacles and completes the distinction of three kinds of obstacles with the success rate of the algorithm has reached 94.29%. It can be seen from Table 1 that the detection errors of the length and width of the obstacles are 0.1 m and 0.18 m, the error percentage is 2.5% and 9%, respectively. The detection accuracy of the height is 0.003 m, which reaches the millimeter level and 0.6% error rate. The average detection error of the algorithm for negative obstacle depth and trench width is 0.020 m and 0.039 m. While completing the above detection, the algorithm also maintains excellent real-time performance, with an average detection time of 16.8 ms. This is due to the low noise of the simulated environment and the result of the ideal obstacle scene.

#### 4.2 Validated in indoor environment

This system is also verified experimentally in an indoor real environment. We created multiple obstacle environments with cardboard boxes, wooden boards, etc. Algorithms are validated on tracked vehicles. Figure 8 (top) shows the built indoor obstacle environment, and Figure 8 (bottom) shows the elevation map and obstacle detection results. Consistent with the simulation results, our work can accurately detect the type of each obstacle. Results of indoor experiments are shown in Table 2. In the real environment, the error of the system for obstacle size detection increases slightly. The accuracy of the detection algorithm is still excellent, reaching 96.5%. Due to the influence of sensor noise and environment, the detection accuracy (3.6%–6.5%) and time (21.5 ms) is not as good as the simulation result.

#### 4.3 Validated in outdoor environment

The system is finally verified by experiments in an outdoor real environment. Corresponding to the obstacle type, we selected three scenarios for outdoor verification experiments. The algorithm was verified on a four-wheeled vehicle. Figure 9 (top) shows the outdoor obstacle environment, and Figure 9 (bottom) shows the elevation map and obstacle detection results. The results are shown in Table 3. Due to the complexity of the outdoor environment and the irregularity of obstacles, the detection error (4.0%–6.6%) and time (22.2 ms) slightly increases compared to the simulation and indoor experiments. However, the algorithm has completed multiple identifications of obstacles with a high success rate (95%).

**Figure 7** Verified in a simulation environment

**Notes:** The obstacles detected in (a)–(e) are: regular obstacles, irregular obstacles and negative obstacle, trench obstacle, complex obstacles and inclined arrangement obstacles, respectively. The green box is the obstacle detection area

**Source:** Authors' own work

## 5. Conclusion and future work

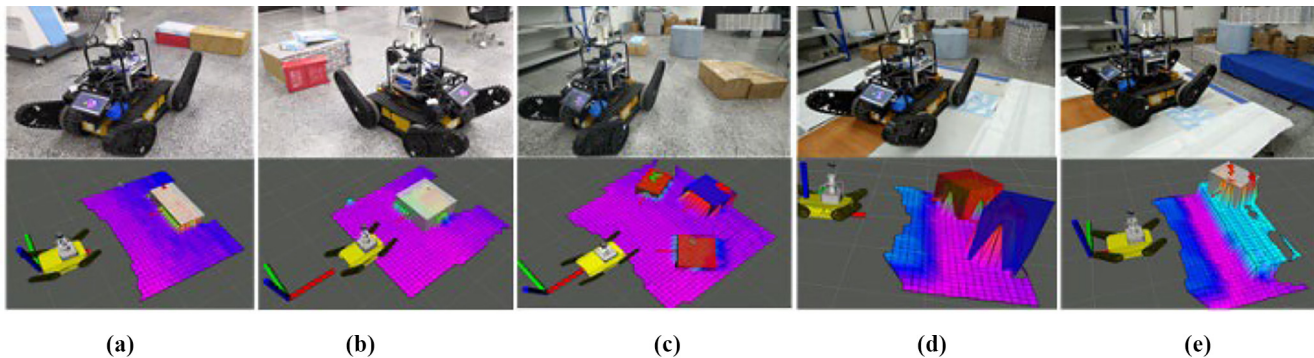
This paper presents an obstacle detection system based on elevation maps capable of detecting positive obstacles, negative obstacles and trench obstacles in various environments. The

paper initially introduces the elevation mapping algorithm based on the Grid Map Library. The mapping algorithm takes the measurement noise of sensors and the localization uncertainties of robot into account. The mapping is realized by two steps: measurement update and movement update. Before identifying obstacles, the scheme completes the ground segmentation and obstacle clustering based on plane fitting algorithm and the idea of Euclidean clustering. For positive obstacles, an algorithm based on the smallest rectangular bounding box is proposed. For negative obstacles and trench obstacles, the algorithm is conducted based on the feature of data absence in the elevation map so as to initially locate them, while final types are determined in accordance with further detection. The proposed detection system is verified in Gazebo simulation environment, real indoor environment and real outdoor environment. Experiments show that in a real outdoor environment, the algorithm efficiently and

**Table 1** Results of simulation experiment

| Type     | Error size (m)         | Error (%)     | Success rate | Time (ms) |
|----------|------------------------|---------------|--------------|-----------|
| Positive | 0.100, 0.180 and 0.003 | 2.5, 9.0, 0.6 | 23/25        | 17.4      |
| Negative | 0.020                  | 2.2           | 5/5          | 15.1      |
| Trench   | 0.039                  | 3.9           | 5/5          | 15.8      |
| Average  | –                      | –             | 94.29%       | 16.8      |

**Source:** Authors' own work

**Figure 8** Verified in an indoor environment

**Notes:** The scene contains several positive obstacles, one negative obstacle and one trench obstacle. The characteristics of obstacles detected in scenarios (a)–(e) are: regular positive obstacles, complex positive obstacles, multiple positive obstacles, negative obstacles and positive obstacles, trenches and positive obstacles, respectively

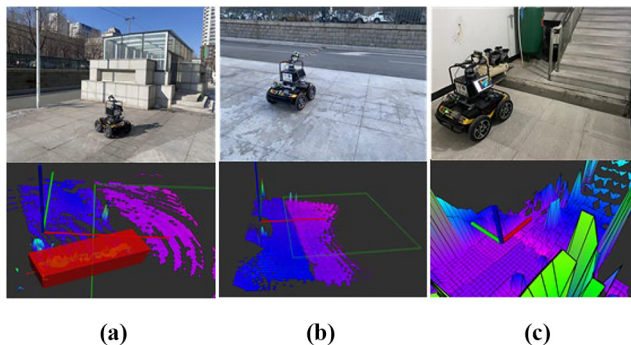
**Source:** Authors' own work

Table 2 Results of indoor experiment

| Type     | Error size (m)         | Error (%)     | Success rate | Time (ms) |
|----------|------------------------|---------------|--------------|-----------|
| Positive | 0.043, 0.040 and 0.019 | 5.3, 5.3, 6.3 | 39/40        | 22.6      |
| Negative | 0.018                  | 3.6           | 20/20        | 19.9      |
| trench   | 0.059                  | 6.5           | 18/20        | 21.1      |
| Average  | –                      | –             | 96.5%        | 21.5      |

Source: Authors' own work

Figure 9 Verified in an outdoor environment



Notes: The types of obstacles represented by (a)–(c) are: irregular positive obstacle, negative obstacle and trench obstacle  
Source: Authors' own work

Table 3 Results of outdoor experiment

| Type     | Error size (m)         | Error (%)     | Success rate | Time (ms) |
|----------|------------------------|---------------|--------------|-----------|
| Positive | 0.352, 0.203 and 0.162 | 5.5, 5.8, 6.5 | 38/40        | 23.4      |
| Negative | 0.008                  | 4.0           | 19/20        | 20.3      |
| Trench   | 0.026                  | 6.6           | 19/20        | 21.6      |
| Average  | –                      | –             | 95.0%        | 22.2      |

Source: Authors' own work

accurately completes the obstacle detection task with an average time of 22.2 ms and a success rate of 95%. The error range of obstacle size detection is 4%–6.6%, which meets the next step obstacle-crossing requirements.

Future work will involve the optimization of the algorithm to improve efficiency, as well as the improvement of the algorithm's functions to detect negative obstacles and trench obstacles to expand its applicability and meet more needs stemming from different scenarios.

## References

- Arnon, D.S. and Gieselmann, J.P. (1983), "A linear time algorithm for the minimum area rectangle enclosing a convex polygon".  
Bai, H., Du, Z., Zhu, H., Ding, P., Wang, G., Wang, H., Xu, W. and Wang, W. (2023a), "Multi-source term estimation based on parallel particle filtering and dynamic state space in

unknown radiation environments", *Building and Environment*, Vol. 236, p. 110281.

- Bai, H., Gao, W., Ma, H., Ding, P., Wang, G., Xu, W., Wang, W. and Du, Z. (2023b), "A study of robotic search strategy for multi-radiation sources in unknown environments", *Robotics and Autonomous Systems*, Vol. 169, p. 104529.  
Belter, D. and Skrzypczyński, P. (2011), "Rough terrain mapping and classification for foothold selection in a walking robot", *Journal of Field Robotics*, Vol. 28 No. 4, pp. 497–528.  
Belter, D., Łabcki, P. and Skrzypczyński, P. (2012), "Estimating terrain elevation maps from sparse and uncertain multi-sensor data", *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 715–722.  
Chu, P., Cho, S., Sim, S., Kwak, K. and Cho, K. (2017), "A fast ground segmentation method for 3D point cloud", *J. Inf. Process. Syst*, Vol. 13 No. 3, pp. 491–499.  
Dornhege, C. and Kleiner, A. (2007), "Behavior maps for online planning of obstacle negotiation and climbing on rough terrain", *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3005–3011.  
Drulea, M., Vataavu, A., Mandici, S. and Nedevschi, S. (2016), "An omnidirectional stereo system for logistic plants. Part 2: stereo reconstruction and obstacle detection using digital elevation maps", *In Proc. Rom. Acad.*  
Fankhauser, P. and Hutter, M. (2016), "A universal grid map library: implementation and use case for rough terrain navigation", *Robot Operating System (ROS) the Complete Reference*, Vol. 1, pp. 99–120.  
Fankhauser, P., Bloesch, M. and Hutter, M. (2018), "Probabilistic terrain mapping for mobile robots with uncertain localization", *IEEE Robotics and Automation Letters*, Vol. 3 No. 4, pp. 3019–3026.  
Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M. and Siegwart, R. (2014), "Robot-centric elevation mapping with uncertainty estimates", *Mobile Service Robotics*, pp. 433–440.  
Florin, O., Nedevschi, S., Meinecke, M.-M. and To, T.-B. (2007), "Road surface and obstacle detection based on elevation maps from dense stereo", *2007 IEEE Intelligent Transportation Systems Conference*, pp. 859–865.  
Forouher, D., Besselmann, M.G. and Maehle, E. (2016), "Sensor fusion of depth camera and ultrasound data for obstacle detection and robot navigation", *2016 14th international conference on control automation robotics and vision (ICARCV)*, pp. 1–6.  
Gao, H., Qin, Y., Hu, C., Liu, Y. and Li, K. (2021a), "An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario", *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12.  
Gao, H., Su, H., Cai, Y., Wu, R., Hao, Z., Xu, Y., Wu, W., Wang, J., Li, Z. and Kan, Z. (2021b), "Trajectory prediction of cyclist based on dynamic bayesian network and long short-term memory model at unsignalized intersections", *Science China Information Sciences*, Vol. 64 No. 7, p. 172207.  
Gao, H., Kan, Z. and Li, K. (2022), "Robust lateral trajectory following control of unmanned vehicle based on model

- predictive control”, *IEEE/ASME Transactions on Mechatronics*, Vol. 27 No. 3, pp. 1278-1287.
- Graham, R.L. (1972), “An efficient algorithm for determining the convex hull of a finite planar set”, *Information Processing Letters*, Vol. 1 No. 4, pp. 132-133.
- Hou, J.F.J. (1999), *Clustering with Obstacle Entities*, Simon Fraser University.
- Hu, Z. and Uchimura, K. (2005), “UV-disparity: an efficient algorithm for stereovision based scene analysis”, *IEEE Proceedings. Intelligent Vehicles Symposium 2005*, pp. 48-54.
- Kleiner, A. and Dornhege, C. (2007), “Real-time localization and elevation mapping within urban search and rescue scenarios”, *Journal of Field Robotics*, Vol. 24 Nos 8/9, pp. 723-745.
- Labayrade, R., Aubert, D. and Tarel, J.P. (2002), “Real time obstacle detection in stereovision on non flat road geometry through” v-disparity” representation”, *Intelligent Vehicle Symposium*, Vol. 2, pp. 646-651.
- Larson, J. and Trivedi, M. (2011), “Lidar based off-road negative obstacle detection and analysis”, *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 192-197.
- Li, C., Li, L., Wu, G., Xu, N.-Y., Gao, H. and Guo-Wen, T. (2016), “A visual 3D modelling system for the robot autonomous obstacle negotiation in substation”, *2016 4th International Conference on Applied Robotics for the Power Industry (CARPI)*, pp. 1-4.
- Mancini, M., Costante, G., Valigi, P. and Ciarfuglia, T.A. (2018), “J-MOD2: joint monocular obstacle detection and depth estimation”, *IEEE Robotics and Automation Letters*, Vol. 3 No. 3, pp. 1490-1497.
- Moosmann, F., Pink, O. and Stiller, C. (2009), “Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion”, *2009 IEEE Intelligent Vehicles Symposium*, pp. 215-220.
- Naujoks, B. and Wuensche, H.-J. (2018), “An orientation corrected bounding box fit based on the convex hull under real time constraints”, *IEEE Intelligent Vehicles Symposium*, pp. 1-6.
- Ortigosa, N., Morillas, S. and Peris-Fajarnés, G. (2011), “Obstacle-free pathway detection by means of depth maps”, *Journal of Intelligent & Robotic Systems*, Vol. 63 No. 1, pp. 115-129.
- Shang, E., An, X., Li, J. and He, H. (2014), “A novel setup method of 3D LIDAR for negative obstacle detection in field environment”, *17th International IEEE Conference on Intelligent Transportation Systems*, pp. 1436-1441.
- Shi, Y., Dong, W., Lin, W. and Gao, Y. (2022), “Soft wearable robots: development status and technical challenges”, *Sensors*, Vol. 22 No. 19, p. 7584.
- Sokolov, M., Afanasyev, I., Klimchik, A. and Mavridis, N. (2017), “HyperNEAT-based flipper control for a crawler robot motion in 3D simulation environment”, *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2652-2656.
- Tian, Y., Song, W., Chen, L., Sung, Y., Kwak, J. and Sun, S. (2020), “A fast spatial clustering method for sparse LiDAR point clouds using GPU programming”, *Sensors*, Vol. 20 No. 8, p. 2309.
- Vincent, I. and Sun, Q. (2012), “A combined reactive and reinforcement learning controller for an autonomous tracked vehicle”, *Robotics and Autonomous Systems*, Vol. 60 No. 4, pp. 599-608.
- Wang, G., Wang, W., Ding, P., Liu, Y., Wang, H., Zhong, F., Bai, H., Zhu, H. and Du, Z. (2023), “Development of a search and rescue robot system for the underground building environment”, *Journal of Field Robotics*, Vol. 40 No. 3, pp. 655-683.
- Wang, J., Song, Q., Jiang, Z. and Zhou, Z. (2016), “A novel InSAR based off-road positive and negative obstacle detection technique for unmanned ground vehicle”, *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1174-1177.
- Wermelinger, M., Fankhauser, P., Diethelm, R., Krusi, P., Siegwart, R. and Hutter, M. (2016), “Navigation planning for legged robots in challenging terrain”, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1184-1189.
- Xie, G., Gao, H., Qian, L., Huang, B., Li, K. and Wang, J. (2018), “Vehicle trajectory prediction by integrating physics- and Maneuver-based approaches using interactive multiple models”, *IEEE Transactions on Industrial Electronics*, Vol. 65 No. 7, pp. 5999-6008.
- Xu, Y., Cao, J., Jia, P. and Zhang, Z. (2012), “The research of obstacle detection based on AK-means clustering algorithm in crosscountry”, *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Vol. 3, pp. 1196-1199.
- Ye, C. and Borenstein, J. (2003), “A new terrain mapping method for mobile robots obstacle negotiation”, In *Unmanned ground vehicle technology V*, Vol. 5083, pp. 52-62.
- Ye, C. and Borenstein, J. (2004), “A novel filter for terrain mapping with laser rangefinders”, *IEEE Transactions on Robotics*, Vol. 20 No. 5, pp. 913-923.
- Yuan, X., Tang, X., Xue, S. and Zhao, C. (2015), “Obstacle detection based on image and laser points fusion for a small ground robot”, *2015 IEEE International Conference on Information and Automation*, pp. 386-391.
- Zaiane, O.R. and Lee, C.H. (2002), “Clustering spatial data in the presence of obstacles: a density-based approach”, *Proceedings International Database Engineering and Applications Symposium*, pp. 214-223.
- Zermas, D., Izzat, I. and Papanikolopoulos, N. (2017), “Fast segmentation of 3d point clouds: a paradigm on lidar data for autonomous vehicle applications”, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5067-5073.
- Zhu, H., Cao, C., Xia, Y., Scherer, S., Zhang, J. and Wang, W. (2021), “DSVP: dual-stage viewpoint planner for rapid exploration by dynamic expansion”, *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7623-7630.

## Further reading

- Hongbiao, Z., Liu, Y., Wang, W. and Du, Z. (2021), “Obstacle negotiation analysis of track-legged robot based on

terramechanics”, *Industrial Robot: The International Journal of Robotics Research and Application*, Vol. 48 No. 6, pp. 812-822.

### About the authors

**Han Wang** received the BSc degree from the School of Transportation, Northeast Forestry University in 2019, and the MEng degree from the School of Automotive Engineering, Harbin Institute of Technology in 2021, where he is currently pursuing the PhD degree with the School of Mechatronics Engineering. His research interests include obstacle detection, drivable area detection and autonomous navigation in uneven environment.

**Quan Zhang** obtained his MSc in Mechatronic Engineering from Harbin Institute of Technology in 2020. His research interests lie in ground robots and SLAM.

**Zhenquan Fan** obtained his BSc in Mechatronic Engineering from Harbin Institute of Technology at Weihai in 2019. Currently, he is a master’s student at Harbin Institute of Technology. His research interests lie in ground robots and autonomous.

**Gongcheng Wang** obtained his MSc in Mechatronic Engineering from Shandong University in 2020. At present, he is a PhD student at Harbin Institute of Technology, pursuing his study in State Key Laboratory of Robotics and System. His main research interests lie in vehicle/arm coordination and remote operation.

**Pengchao Ding** obtained his MSc in Mechatronic Engineering from Harbin Institute of Technology in 2020. Currently, he is a PhD student at Harbin Institute of Technology, pursuing his study in State Key Laboratory of Robotics and System. His main research interests lie in autonomous exploration.

**Weidong Wang** received the BS, MS and PhD degrees in Mechatronic Engineering from Harbin Institute of Technology, in 2002, 2004 and 2009, respectively, where he is currently with the State Key Laboratory of Robotics and System. His research expertise is in the general areas of robotics and mechatronics. His research interests include motion planning, computer vision, human-machine interaction and its application in field robot and surgical robot. Weidong Wang is the corresponding author and can be contacted at: [wangweidong@hit.edu.cn](mailto:wangweidong@hit.edu.cn)