Abstract— Analysing the traversability from the point cloud input is a critical task for mobile robot navigation. However, there is no mature method for mobile robots to exploring complicated 3D environments, especially those with multi-layered roads. This paper proposes a new traversability map based on the point cloud input, and the map will help analyze where the mobile robot can move in the whole space. The proposed 3D traversability map consists of nodes and edges generated according to the mobile robot’s geometry and driving performance. Nodes represent where the robot can be placed, and edges indicate the cost of moving to another node. The traversability map is stored in digraph form and is convenient to be applied to path planning. Experiments show the successful results of traversability map generation and path planning with D algorithm in different situations.

I. INTRODUCTION

Mobile robot has been applied in unknown environment exploration for many years. With the fusion of various sensors, like LIDAR and cameras, the surrounding environment can be reconstructed in a point cloud map. However, the traversability in the whole space cannot be identified with merely a point cloud map for mobile robots. Thus, a kind of traversability map is necessary for exploration tasks, and this is still a challenging task, especially in complicated environments with obstacles both on the ground and overhanging.

Currently, challenges in 3D traversability map construction include:

1. Identification of the path for mobile robots under multi-layered roads. Although 3D point cloud map can show the position of objects in the environment, it cannot indicate which are obstacles and which are roads. Especially when there are multi-layered roads, like the bridge structure, many traversability maps failed.

2. Inconsistency of traversability for different kinds of mobile robots. Whether a mobile robot can traverse smoothly in a given environment depends on the geometry structure and driving performance. Obstacles that stop a robot in small size may not stop bigger ones.

Aiming at the two challenges above, this paper proposes a new kind of 3D traversability map. The point cloud map and the mobile robot’s size are both considered, and the data structure of graph is applied to store the traversability information of the whole 3D environment. The graph generated can also be used in the task of path planning and navigation for mobile robots.

To get the roads that mobile robots can be placed on, a set of discrete nodes are generated to represent the positions that mobile robots can be placed. These nodes are distributed over the whole space from the point cloud map and extracted by the distribution situation of points around them. The traversability between nodes is represented by directed edges and quantified for application in path planning.

To deal with the inconsistency of traversability for different robots, mobile robots’ critical parameters are extracted. This paper considers mobile robots with four wheels. Thus, the four-wheeled robot’s geometry shape and driving performance are parameterized and applied into the generation of 3D traversability map.

II. RELATED WORK

A. Maps for Mobile Robot

The construction of 3D traversability map needs to consider both the environment and the state of the robot. At present, the most popular maps are occupied grid map and elevation map. Occupied grid maps contain 2D and 3D maps, showing the position of obstacles. Besides, scholars extended occupied maps to improve computational efficiency [1]. Elevation map presents the geometry information of the surface of the ground. In order to represent the terrain of different layers, scholars proposed multi-level elevation map [2,3]. In addition, there are other forms of maps representing multi-layer environments [4].

With the maps mentioned above, traversability maps are generated, considering the state of mobile robots. Sock [5] built a 2D traversability map from estimating the terrain with LIDAR and camera. Mongus [6] generated a traversability map from digital terrain models. Zhao [7] proposed a semantic probabilistic traversable map for navigation tasks. Tang [8] proposed a traversability map in 2D form from an elevation map. However, most traversability maps are established based on 2D maps. Thus they are not easy to directly describe the traversability of mobile robots in the whole 3D space.

B. Traversability Analysis

For a mobile robot, identifying the traversability of roads is a significant task. The inevitable problems are whether mobile robots can go through and the cost consumed when traversing on.

Recently, there are two types of methods for these problems, one uses classifiers, the other uses learning algorithms. Many researchers learned traversability from geometry [9]. Plaza-Leiva [10] got the ground’s traversability with gradient information and unevenness of the ground surface. Reddy [12] judged the traversability from unevenness using LIDAR. Other

In order to evaluate the cost to traverse on various roads, most researchers set weights for different parameters on mobile robot’s traversability. However, for mobile robots, not only downside condition affects the traversability, but also the upside environment may cause a collision. When it comes to upside condition, merely the height size of mobile robots is considered. Current methods usually work on flat roads but will fail on uneven terrain and multi-layered situation.

C. Topological Graph

Some scholars applied a topological graph to express the traversability of mobile robots. In different researches, nodes and edges in topological graphs have different meanings. Blöchli [16] and Gomez [17] defined nodes as locations that robots can reach in space. They connected nodes accessible to each other with edges and used sparse topology graphs to reduce data storage and to improve computational efficiency. Nijjima [18] used nodes to represent landmarks in urban environments and applied topology to describe the situation of the complex environment intuitively. T. Ohradzansky [19] proposed an unknown environment exploration method based on bio-inspired reactive control and metric-topological planning. Ramaithitima [20] used nodes to represent swarm robots and probed the environment through the distribution of the robot cluster.

III. THE PROPOSED METHOD

For mobile robots that traverse complicated environments, a map representing the whole space’s traversability will help greatly. The traversability map proposed in this paper is defined as a graph consisting of nodes and edges. Nodes indicate the positions where mobile robots could be placed, and edges between two nodes indicate the difficulty that the mobile robot moves from one node to another. The traversability map is generated from a point cloud map and the mobile robot’s parameters with the process shown in Fig. 1.

A. Node Selection

With the point cloud map input, the first step is to find all the possible positions that the mobile robot could be placed stably. \( W \) denotes the coordinate system of the point cloud map. Space is discretized in three directions of \( x_w, y_w \) and \( z_w \) with resolutions of \( r_x, r_y \) and \( r_z \). Each discrete point is defined as a node where the mobile robot will be placed.

Whether a node is satisfied depends on the distribution of points around it. When a mobile robot is placed in a node’s position, points under the node are regarded as hard objects that could hold the mobile robot on, while points above are seen as obstacles that stop the mobile robot from being placed here.

In this paper, the traversability of four-wheeled mobile robots is considered. The model of mobile robot is simplified as shown in Fig. 2. Critical geometry parameters are extracted for the analysis of contact with the point cloud. The length of the mobile robot in the forward direction is defined as \( L_f \), and the width is defined as \( L_w \). Each wheel is considered as a sphere whose radius is defined as \( r \). The height robot is \( H_s \) and the height of the chassis is \( H_c \). Besides, the climbing ability of the robot is parameterized as the maximum climbing angle \( \alpha \). The center of the robot coordinate \( R \) is set on the projection of the robot center.

The coordinate system \( N \) is built in the position of each node. \( P \) denotes all the points in the point cloud map. \( P_w \{x_w, y_w, z_w\} \) is the corresponding coordinate in \( W \) and \( P_N \{x_N, y_N, z_N\} \) is the corresponding coordinate in \( N \). The relationship between \( P_w \) and \( P_N \) is shown in Eq. (1). \( O_w^N \) denotes the coordinate of the node in \( W \).

\[
P_w = O_w^N + P_N
\]

The space around a node is divided into five parts, named \( U, A, B, C, \) and \( D \). As shown in Fig. 3, points in space \( A, B, C, \) and \( D \) could hold the mobile robot, while points in space \( U \) may cause a collision. When there is no point in space \( U \) and there exist points in all of \( A, B, C, \) and \( D \), this node will be extracted for further computation. \( \delta h \) is less than \( r \) to avoid missing satisfied nodes.

All directions should be taken into consideration for a node. But with the balance of computation cost, we only consider the condition when yaw angle changes along \( z_w \). Besides, considered the robot climbing ability, the space of \( U \) should be modified for the slope situation in Fig. 4. Eventually, points that satisfy Eq. (2) are considered in space \( U \), while points that satisfy Eq. (3) are considered in space \( A \). The space \( B, C, \) and \( D \) can be calculated by mirror symmetry.
Figure 2. Model of mobile robot.

Figure 3. Divided space around the node.

Figure 4. Modified space U based on the angle.

\[
\begin{align*}
&\begin{cases} x_n^2 + y_n^2 \leq (L_1/2)^2 + (L_2/2)^2 \\
\tan \alpha \sqrt{x_n^2 + y_n^2} < z_n \leq H \end{cases} \\
&\begin{cases} x_n^2 + y_n^2 \leq (L_1/2)^2 + (L_2/2)^2 \\
-\delta h < z_n \leq \tan \alpha \sqrt{x_n^2 + y_n^2} \end{cases} \\
x_n \geq 0 \\
y_n \geq 0
\end{align*}
\]

The screening method above extracts nodes that are suitable for the robot to be placed. The result is shown in Fig. 5. For most situations, the extraction method will greatly reduce the number of nodes. So the cost of further computation is at 2D level, or only several times in the multi-layered environment.

B. Pose Estimation

After initially extracting nodes from points around, the further selection is applied according to the pose of the mobile robot when put in each node’s position. The pose estimation method proposed refers to the manner in [8] and extends the situation to 3D.

First of all, points below each node are selected according to the projection of wheels. As shown in Fig. 6, points are clustered according to the wheels, as the cluster \(C_A, C_B, C_C\) and \(C_D\). When there are no points under each wheel, this node is eliminated.

In a 3D situation, there exist points covered by other points in \(z\). The covered points have no chance to contact wheels. So it is necessary to extract points on the surface for further calculation.

According to the analysis in [21], there are two poses for the mobile robot placed in the same node, as \(n_h^1\) and \(n_h^2\). Every cluster’s average value is computed to get which cluster to select with Algorithm 1.

**Algorithm 1 Compute normal \(n_h^1\) and \(n_h^2\)**

1. Input points of the cluster \(C_A, C_B, C_C\) and \(C_D\)
2. Compute \(z_w^1, z_w^2, z_w^3\) and \(z_w^4\) with Eq. (4);
3. if \(z_w^4 > z_w^1 + z_w^3\) with Eq. (4);
4. Compute \(n_h^1\) with \(C_A, C_B\) and \(C_C\);
5. Compute \(n_h^2\) with \(C_A, C_B, C_C\) and \(C_D\);
6. else
7. Compute \(n_h^1\) with \(C_A, C_B\) and \(C_D\);
8. Compute \(n_h^2\) with \(C_B, C_C\) and \(C_D\);

\[
z_w^i = \frac{1}{k} \sum_{j=1}^{k} z_w^j, i = A, B, C, D
\]

\(k_i\) is the number of points in the cluster \(C_i\). \(z_w^i\) is the average height of points in \(C_i(i = A, B, C, D)\).

The normal of points cluster \(n[n_1, n_2, n_3](n_i > 0)\) can be obtained from PCA (Principal Component Analysis), and the center of the cluster is \(p_{\tau} = \{x_{\tau}, y_{\tau}, z_{\tau}\}\). With the result of the normal computed, the position \(R_{\tau} = \{x_{\tau}, y_{\tau}, z_{\tau}\}\) of the mobile robot can be calculated according to Eq. (5). Then the position of the node is translated from \(O_{n}\) to \(O_{h}\) in Fig. 7.
When the inclination angle is larger than $\alpha$, the robot is not able to be placed in this node. Another condition that stops the robot from being placed here is that points under the robot are higher than the chassis, shown in Fig. 8. When there are points around satisfied Eq. (6), this node is defined as unreachable.

$$\langle P_w - O^p_k \rangle n > H_c$$

(6)

After the process mentioned above, all reachable nodes are extracted. For more accuracy of pose estimation, the process of selecting points under wheels and computing pose can be performed multi-times. The mobile robot’s yaw angle of the mobile robot can be changed to estimate the pose in different directions.

C. Traversability Analysis

With the reachable nodes extracted, where the mobile robot can be placed is shown. Then the edge of the graph will be calculated as the traversability of 3D space.

First of all, the connectivity of nodes will be computed. Each edge links two nodes that are closed enough and with a low gradient. The short distance ensures there is no unreachable area between two nodes, and the low gradient ensures the mobile robot has enough engine to drive up or down. Edges between nodes show the connectivity of the whole map for the mobile robot.

Then, the difficulty for the mobile robot to traverse from one node to another is defined as each edge’s weight. The weight depends on the distance between two nodes and the distribution of points under two nodes. The weight $w$ is defined as Eq. (7). $l$ denotes the distance between two nodes, $w_r$ denotes the roughness of point cloud between two nodes and $w_s$ denotes the slope degree. $k_r$ and $k_s$ denote the ratio of $w_r$ and $w_s$ according to the condition of the real situation.

$$w = l(k_rw_r + k_sw_s)$$

(7)

IV. SIMULATION AND EXPERIMENT

In this section, we will evaluate the performance of the traversability map proposed. Besides, path planning is performed in the map generated. Experiments are performed on a computer equipped with an Ubuntu 18.04 64bit operation system, an Intel core i5-8400 CPU and 8GB RAM.

There are three point cloud datasets selected to test the performance of traversability map generation. Dataset 1 is generated from a simulator. It has an obstacle like a bridge that the mobile robot can climb up the slope and move through the hole. It also contains a convex obstacle flat enough for the robot to be placed but high enough not to travel up and down.

Dataset 2 is a subset of the point cloud from SmartGeoMetrics. The part selected is a typical undulating terrain with two layers. Dataset 3 is from Freiburg Campus. It is a vast outdoor scenario from 3D scans.

A. Mapping performance

The performance of 3D traversability map generation in three datasets is shown in Fig. 9. Since the evaluation of a spatial map representation is still an open problem, we will evaluate the performance of maps qualitatively.

In Dataset 1, two clusters of nodes are shown in different colors in Fig. 9 (b). Nodes in the same color are linked with edges, indicating the traversability among nodes. Nodes in yellow have no edges linked to nodes on the ground, which is the same as the environment simulated. Besides, nodes in cyan contain two layers, the same as the traversability of the obstacle bridge. In Dataset 2, Fig. 9 (d) shows the waves of nodes on the rugged terrain, and it is consistent with the points distribution. The traversability of the robot on two layers is presented in two kinds of colored nodes. In Dataset 3, Fig. 9 (f) presents the nodes and edges in the outdoor environment. Most nodes extracted are colored in the same color, showing that the mobile robot’s vast traversable areas.

Above all, the traversability map proposed can show the 3D traversability of the mobile robot. The traversability map proposed in [8] can only be applied in 2.5D environment with terrain map. It will not work in the multi-layered environment. The 2.5D NDT map proposed in [4] can represent whole space, but without the geometry message of mobile robot input, and collision conditions cannot be avoided.

B. Path Planning

With the traversability map generated, path planning is easy to perform. The data structure of the map is a digraph with nodes and directed edges. Nodes store the robot’s position when placed on and traversability message like whether a collision will happen. Directed edges store the difficulty for the mobile to


2http://ais.informatik.uni-freiburg.de/projects/datasets/fr360/freiburgCampu
s360_3D.zip
traverse from one node to another. Thus the path planning of robot can be converted to the traditional problem of finding shortest path and solved in many classical algorithms like Breadth/Depth First Algorithm, Dijkstra Algorithm and Bellman-Ford Algorithm. In this paper, we apply Dijkstra Algorithm and test the performance in path planning.

The path planning results are shown in Fig. 10, with the yellow line indicating the path in point cloud map and traversability map, respectively. We can see that the path planning algorithm works well in all three situations. It is worth noting that the start point and goal point are in different layers in Dataset 1; this indicates that the proposed traversability map works in multi-layered environments. The time cost of the proposed method is shown in Table I.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Points number</th>
<th>Size of map (m)</th>
<th>Map generation(s)</th>
<th>Path planning(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>175k</td>
<td>26<em>12</em>5</td>
<td>0.874</td>
<td>0.014</td>
</tr>
<tr>
<td>2</td>
<td>231k</td>
<td>20<em>16</em>6</td>
<td>1.564</td>
<td>0.019</td>
</tr>
<tr>
<td>3</td>
<td>760k</td>
<td>100<em>100</em>10</td>
<td>5.857</td>
<td>0.079</td>
</tr>
</tbody>
</table>

a. Size of mobile robot: $l_x = l_y = l_z = 1.0 m, h_x = 0.3 m, k_x = 0.3 m, a_x = 0.0 m/s, a_x = 30^\circ$

b. Resolution of node discretization in dataset 1 and dataset 2: $r = 0.3 m, r_y = 0.5 m, r_z = 0.2 m$

c. Resolution of node discretization in dataset 3: $r = 1.0 m, r_y = 1.0 m, r_z = 0.2 m$

V. CONCLUSION

This paper proposes a 3D traversability map generated from point cloud map. It indicates where mobile robot can be placed and where robot can move in the whole space. The proposed traversability map consists of nodes and edges, representing the reachable place and the cost of traverse, respectively. When given point cloud map, the reachable areas are extracted discretely according to the mobile robot’s geometry size and driving performance. Then the cost of travel between nodes is computed by the distribution of points around. Besides, the two poses of four-wheeled in every node can be calculated with the PCA algorithm. With the graph generated, D Algorithm is applied to find the optimal path. Experiments show that the proposed traversability map works well in the complicated, multi-layered environment, and can be easily applied to path planning tasks.

Future work will focus on applying the 3D traversability map in physical mobile robots and using the map in navigation tasks.

REFERENCES


Figure 9. Traversability maps generation. (a), (c) and (e) are point cloud of Dataset 1, 2 and 3 respectively. (b), (d) and (f) are traversability maps generated from Dataset 1, 2 and 3 respectively.

Figure 10. Path planning results. (a), (c) and (e) are point cloud with the path planned of Dataset 1, 2 and 3 respectively. (b), (d) and (f) are traversability map with the path planned of Dataset 1, 2 and 3 respectively.