Industrial Project: Driving and Navigation

A project of the 2024 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

Li Pingcong
lipc@shanghaitech.edu.cn

June 25, 2024

Abstract

This project is about a industrial robot. We propose an autonomous driving system using elevation map to help the robot move to the correct place and proceed the industrial mission.

1 Introduction

It is popular now to utilize solar energy to generate electricity. In northwest of China, there is a large amount of flat fields with strong sunlight, which provides great conditions for construction of enormous solar power generating base. However, it's not a simple task to deploy so much giant and heavy solar panels. Therefore, it is proposed that robots could help us complete this tough mission which improves the efficiency and safety. To make the process of construction be fully automatic, we need to develop a control system to control the robot implement the construction.

This project is about to design an autonomous driving system for the robot. In this system, we implement a differential driving controller and a elevation map system for navigation. We take advantages of efficiency of the elevation map to implement a traversability analysis system and a collision detection system. We take every object as a part of the ground and build a new complete ground surface. We implement the traversability analysis and collision detection by calculating the robot's pose according to the elevation date corresponding to each wheel. By doing so, we could reduce memory and time cost for navigation in an unstructured outdoor environment.

2 State of the Art

2.1 papers

The most important task in navigation is collision detection. In order to do this, apparently we need a map. Since our robot runs on a large flat field outdoors and the obstacles are sparse, it is a good choice to use an elevation map which could simplify the task and save time and memory. A paper proposed a universal grid map framework which could help me complete this task through creating an elevation layer. [1]

Certainly if we utilize a map to do collision detection, we have to create the map we need. It is proposed that it's a good way to utilize elevation mapping with a mobile robot which is equipped with a pose estimation and a distance sensor. [2] [3]

This project is based on a industrial robot working in the unstructured outdoor environment. A paper proposed a hybrid map-based path planning system for robot navigation in unstructured environment. [4] The robot in this paper is much smaller than the robot in my project, while the core idea is still inspiring. In this paper, a hybrid map containing a 2D grid map and a 2.5D elevation map is proposed. The 2D grid map provides information about absolutely inpassable areas such as obstacles. The 2.5D elevation map provides more detailed information about the properties of terrain's surface which is used to generate a traversability map for navigation. For those areas with simple obstacles, the 2D grid map is sufficient for navigation. At the same time, it takes much less time and memory. For those areas where the terrian structure is much more complicated, the 2.5D elevation map could help robot learn more about the terrian such as inclination angle and roughness to accurately determine the traversability. To generate this hybrid map, this paper proposed a point cloud segmentation method to filter out the point cloud near the terrian surface and segment the point cloud into grids according to the scale of the robot. By doing so, the map's resolution will suit the requirement of robot's navigation. And it could reduce the impact from some objects like trees or bridges which has possible route under their structures.

2.2 elevation mapping

Mobile robot build on accurate, real-time mapping with onboard range sensors to achieve autonomous navigation over rough terrian. The paper proposes a novel terrian mapping method based on proprioceptive localization from kinematic and inertial measurements only.The proposed method incorporates the drift and uncertainties of the state estimation and a noise model of the distance sensor. It yields a probabilistic terrian estimate as a grid-based elevation map including upper and lower confidence bounds.

The method estimates the elevation map in a robot-centric coordinate frame such that the process of integrating new measurements into the map is only affected by the range sensor noise and the uncertainty of the observable roll and pith angles.The data in the map is updated based on the uncertainty of the incremental motion as the robot moves through the environment. This gives the robot at any point in time an estimate of the terrian from its local perspective. The computation burden of the mapping procedure is lowered by splitting the method into data collection and a map fusion step.

2.3 grid map

This is a C++ library with ROS interface to manage two-dimentional grid maps with multiple data layers. It is designed for mobile robotic mapping to store data such as elevation, variance, color, friction coefficient, foothold quality, surface normal, traversability etc.
It has such features:

1. Multi-layered: Developed for universal 2.5-dimensional grid mapping with support for any number of layers.
2. Efficient map re-positioning: Data storage is implemented as 2D circular buffer. This allows for non-destructive shifting of the map's position without copying data in memory.
3. Based on Eigen: Grid map data is stored as Eigen data types. Users can apply available Eigen algorithms directly to the map data for versatile and efficient data manipulation.
4. Convenience functions: Several helper methods allow for convenient and memory safe cell data access. For example, iterator functions for rectangular, circular, polygonal regions and lines are implemented.
5. ROS interface: Grid maps can be directly converted to and from ROS message types such as PointCloud2, OccupancyGrid, GridCells, and our custom GridMap message.
6. OpenCV interface: Grid maps can be seamlessly converted from and to OpenCV image types to make use of the tools provided by OpenCV.
7. Visualizations: It has plugins to render the map in rviz. In addition, it has a visualization package to visualize the map as point cloud, occupancy grids etc.
8. Filters: It has a framework to process filters on the map. Parsing of mathematical expressions allows to flexibly setup powerful computations such as thresholding, normal vectors, smoothening, variance, inpainting, and matrix kernel convolutions.

3 System Description

3.1 Robot Platform

In this project, the scale of the robot is really large. It is equipped with four pairs of wheels or four tracks. It has a differential driving model. The robot model is shown in figure 1. For security reasons, the model is simplified. The

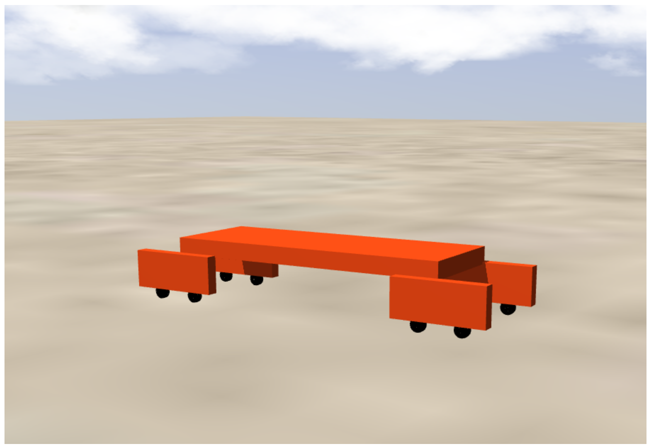

Figure 1: Robot Model

robot has motors to drive the wheels. And it has GPS/dGPS for localization. In perception system, it utilize lidars to scan the environment.

3.2 Methods

We utilize a differential controller from ROS_Control package. We use elevation map as the map representation. In the elevation map, each cell records the heigth of the terrian at this point. In navigation system, we could simultaneously visit the map and calculate the traversability. For traversability analysis, we firstly read the elevation data of robot's wheels and construct a plane according to three higher points.Then we could calculate the pitch angle and roll angle and compare them to the threshold value to determine the tilting condition. Then we randomly select a diagonal line two divide the rectangle of robot's bottom into two triangles. By calculating the angle between normals of these two triangles, we could obtain the stability condition. Two possible unstable state are shown in figure 2 and 3.

For collision detection, traditionally bounding boxes are used to detect collision by checking the overlap of boxes. In this project, in order to simplify the task, we integrate the collision detection into the traversability detection, which means we also utilize elevation map to detect the collision. Since the data of obstacles are also recorded in the elevation map which contains the height of upper surface of the obstacles. So we could assume that the robot stand on a

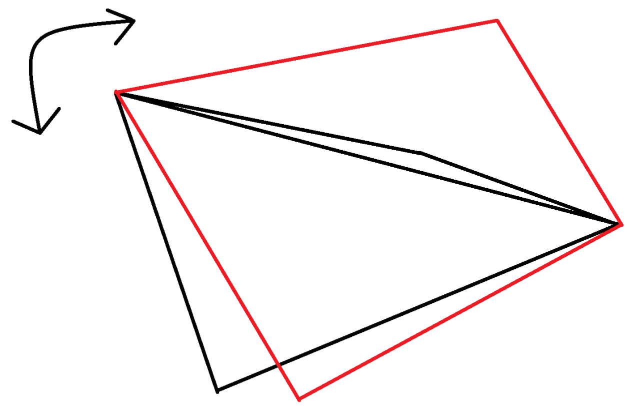

Figure 2: Unstable State 1

spot no matter there is the terrain of some obstacles. Then we could check the tilting condition and stability condition like we did in the traversability analysis to detect the collision between robot's wheels and the obstacles. If a robot's wheel has a collision with an obstacle, then by assuming that the wheel is on the obstacle, we will find the robot is in a bad tilting condition or stability condition since the tilting and stability threshold value is pretty low for the giant industrial robot in this project. For the collision between obstacles and other parts of the robot. We could traverse through the elevation map to check that if the height of the obstacle is over the threshold value. In a simple model, we could compare obstacles' height with a static threshold value since the robot won't drive with a really big pitch angle or roll angle. To make the system more accurate, we could calculate the threshold value of each position of the robot according to the orientation of the robot.

3.3 ROS packages and code

Firstly, we use ROS_Control package for differential control. ROS_Control provides various of motion controllers and a controller management system. We could configure the controller through a yaml file to determine the type, model information, controlling parameters, etc.

Then we use GridMap package to generate and utilize a map containing an elevation layer. This package provides various visualization method to display the map as a 2D map, a 3D mesh, a 3D point could, etc. This package also provides various traversing method for us to visit the map conveniently. We could traverse through a certain area according to a geometry shape. It helps read the data or write the data to generate a map.

In the section of traversability analysis, after receiving the map, we firstly

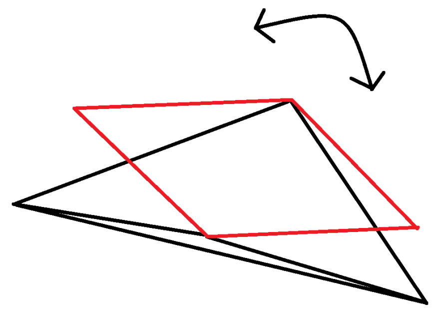

Figure 3: Unstable State 2

calculate the 2D position of the wheels and then read the corresponding elevation value and we will obtain the 3D position of the wheels. Then we sort these points according to the height to find higher 3 points. Then we link the points to generate a plane and calculate the normal vector. By decomposing the normal vector, we will get the pitch angle and the roll angle to determine the tilting condition. Then we select a diagonal line and generate two triangles and calculate their normal vectors to determine the stability condition. After that, according to the model of the robot, we generate 3 rectangles to do the collision detection of robot's left, right and top structures. We cover the map with the rectangles and traverse through each rectangle to compare the elevation data with the collision threshold value. Finally we output all detection information.

4 System Evaluation

We will evaluate the system through simulations. We will create several maps to see if the detection system works. We will run the robot in Gazebo to check the performance of the navigation system. After that we will test the real robot in a testing field. The final goal is that on a certain field for solar panels array construction with specific sparse obstacles such as supporting frames, the robot could drive to the correct place with acceptable errors which means at least the construction system is able to put solar panels on the frames and there is absolutely no collision during the whole process. throught Since this project is based on a real industrial project. The development of our system is limited by the progress of whole project. Currently we have only tested the performance of elevation map. It could do the traversability detection well and display the robot's pose in Rviz as visualization. Here are some images to show the result including map visualization and traversability detection.

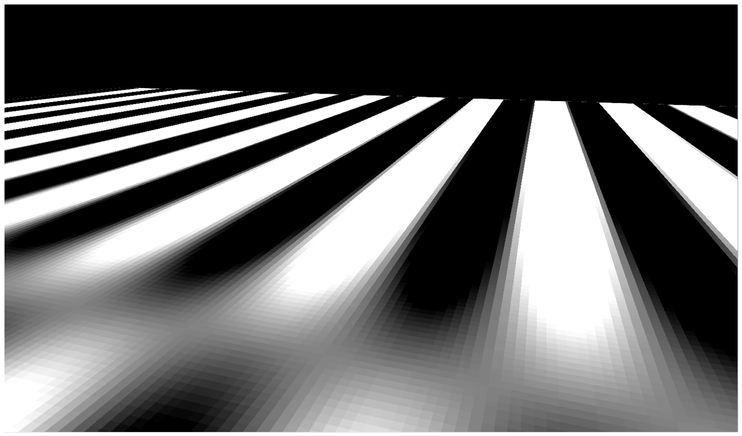

Figure 4: Elevation Map

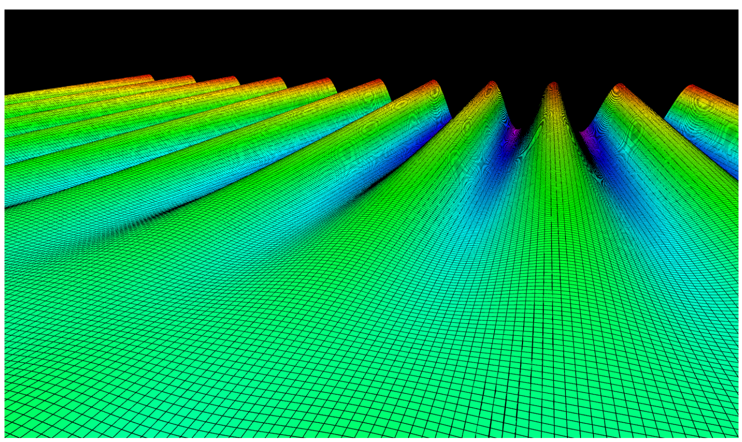

Figure 5: Grid Meshes of the Terrain

5 How To

This project is based on a real industrial project. The hardware belongs to an industrial company. The system is based on ROS1 noetic on Ubuntu 20.04. Due to the security reasons, all codes won't be published.

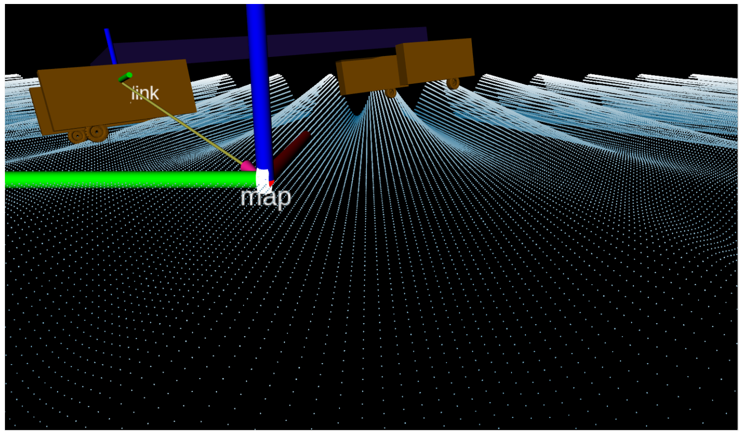

Figure 6: Traversability analysis

6 Conclusions

This project is about developing a driving and navigation system for industrial robot to proceed the mission fully autonomously. We will utilize the elevation map for traversability detection and collision detection. We apply a differential driving model to the four-track robot.

The key contribution in this project is to utilize elevation map for the large-scale industrial robot achieve the traversability analysis and collision detection. It improves the efficiency of the system working for a large robot in outdoor unstructured environment. As a large and heavy robot, on the one hand, it has a much more complicated model than a small one, which requires an accurate system for safety. On the other hand, due to its low flexibility, its actions are simpler and more predictable, which leads to the possibility of using elevation map for navigation. In addition, its low flexibility also makes it feasible to use elevation map for collision detection, which is much more efficient than using collision boxes and doing geometric calculation.

In the future, we will integrate the system into a complete navigation system. And we may have tests through simulation and on a real robot. We may develop a perception system for generating the elevation map according to the data from lidars.

References

[1] Péter Fankhauser and Marco Hutter. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In Anis Koubaa, editor, *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, chapter 5. Springer, 2016.

[2] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3019–3026, 2018.

[3] Péter Fankhauser, Michael Bloesch, Christian Gehring, Marco Hutter, and Roland Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[4] Jiayang Liu, Xieyuanli Chen, Junhao Xiao, Sichao Lin, Zhiqiang Zheng, and Huimin Lu. Hybrid map-based path planning for robot navigation in unstructured environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2216–2223, 2023.