



Obstacle Avoidance System for Autonomous Vehicles

2022 IBM Code Challenge
Team 93

TOC

Overview

Project Objectives

System Architecture

RRT Algorithm

General Structure

Limitations



Overview



The usage of unmanned vehicles is increasing in a variety of fields including but not limited to scientific research, surveillance, agriculture and goods transportation. Along with this, the chances of accidents also increasing drastically. One of the main challenges faced in autonomous vehicle development is

Obstacle avoidance & path planning

Many industries face this problem while incorporating automated vehicles or robots to their work platform .We have developed a prototype as solution for this problem. The vehicle that detects its path by avoiding obstacles in between start point and the destination.



Project Objectives

- 01 Create a working simulation of Obstacle avoidance in autonomous vehicles.
- 02 Developing a path planning mechanism for the demo robot car using the suitable algorithm.
- 03 Successfully implementing the algorithm and achieve a simulated output result for the project.



System Architecture

- The language used to build this project is Python.

- Along with that, we used a python cross platform module (pygame) for the project simulation part.

- The main objective was to find the correct path for the vehicle and for that part we used the RRT (Rapidly exploring Random Tree) algorithm.



Pygame



Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

We created a simulation with the help of pygame , the environment setup consist of an obstacle map image and robot image , we were able to simulate the movement of the robot using RRT algorithm as per our product design.

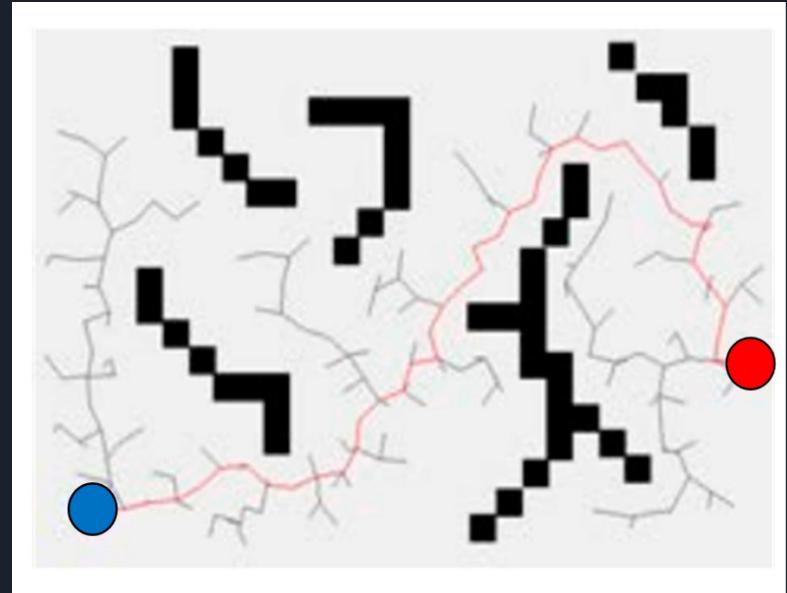


RRT Algorithm

RRT (Rapidly exploring Random Tree) algorithm is effectively used for path planning.

An RRT grows a tree rooted at the starting configuration by using random samples from the search space.

Random nodes are created within the map such that the new tree nodes and edges avoid the obstacle. The tree expands until it reaches the goal.





General structure

- Workspace map as PNG file with free space as white and obstacle space as black (The program can be easily modified to account for other colors. It's only important that there is a distinction between free and obstacle space).
- RRT algorithm for path planning.
- Pygame module is used for simulation.
- Constructed path coordinates are output as a text file.
- Constructed path image is output as PNG file



Limitations

- The program does not calculate the vehicle velocity required to follow the constructed path in real life scenario. It gives only an estimate of the velocity and angular requirements.
- The RRT algorithm assumes the vehicle as a point source. Therefore, the obstacles have to be made larger than it's real life dimensions within the input map to account for the vehicle's width.
- Program can plan paths for static environments only.

Thank you!

Contributors:

Ajay

Nihal

Visakh

