LAB AP - GeoPathPlanner

Group Members

Alessandro Colantuoni (2072461), Umberto Di Tommaso (2133882), Antonio Franzoso (2133047), Paolo Liberti (1787999).

Objective

The following project aims to be a distributed web-application used to find the shortest path between a pair of geographical locations for an aerial (or sea) vehicle avoiding obstacles in a particular searching area. This will be achieved implementing path planning algorithms such that RRT, RRT*, Bug Path, and others.

System description

The project architecture is designed to be distributed as a series of microservices. The front-end one could be a web application having a geojson.io-like map: interacting with it the user will be able to choose waypoints and obstacles (used as system input) in each of the following ways:

- manually select both the waypoints and the obstacles
- manually select waypoints using randomly generated obstacles
- loading waypoints and obstacles from a .geojson file
- loading some pre-defined sets of waypoints and obstacles (stored in a databases)

In addition, the system provides various options to manage the routing request: for example, it could take into account the maximum turning rate of the drone or compare our route to the one that would use a ground vehicle or some other algorithm-based parameters.

Finally, the system computes the optimal route. Each routing request's computation is important to be independent and parallel to the others in order to exploit the distributed nature of it and to optimize the resources. So after each request, the system can scale up or down in order to balance the computational load on it.

When the final route is found, it will be displayed in the map with some options to export it in different formats and save it into a database containing some of the last routing requests, otherwise the failure will be reported.

The back-end programming language will probably be GO or/and Python due to their huge amount of libraries useful to manage geographical features.

The complete list of the microservices could be:

- Front-end service
- Database management
- Path routing service (consists in pre-processing, path finding, post-processing)

Potential users

- Drone's operators
- Delivery system for civilian use
- War / natural disaster zone air search and humanitarian supplies delivery
- Navigation in a sea scenario

Important use cases

The application can be used in air paths routing contexts or for air search on areas of particular interest.
A key use case is the routing between two (or more) points using a flying vehicle and avoiding obstacles. Those obstacles could be real (from a certain database of no-fly-zones or from a set of natural or artificial obstacles) or they could be manually/randomly generated by the user.

In contexts of resource deliveries the application can be used to calculate a route by air and compare it with a terrestrial route, in order to see which mode of transportation is more efficient.

When dealing with areas of particular interest, such as war zones, areas impacted by natural disasters or remote locations, the application can be used to generate air paths useful for humanitarian operators and for supplies deliveries, by taking into account the points to reach and the potential new obstacles present in the areas.

An additional functionality can relate to navigation by sea using aquatic or flying vehicles.