ROS API

1. frontier_explore_node

The frontier explore node takes in the occupancy grid map messages and generates commands for the robot to explore the unknown environment. This exploration node is implemented in the *FrontierExplore* class.

1.1 Actions Called

- move_base (move_base_msgs/MoveBaseAction)
  > move_base actionlib API for posting navigation goals. The target frontier pose is sent to move_base server as a goal to pursue in the world. See move_base API for details.

1.2 Subscribed Topics

- map (nav_msgs/OccupancyGrid)
  > Map which is used for frontier detection and exploration planning. This topic is published by *slam_gmapping* node, the message contains grid data that marks the space as unknown, occupied or open.

1.3 Published Topics

- mobile_base/commands/velocity (geometry_msgs/Twist)

  > The motion control message for turtlebot. The node publish this message to rotate the robot to scan the environment.

- frontier_goal (visualization_msgs/Marker)

  > Marker that visualizes the next frontier target in rviz.

1.4 Services Called

- detect_frontiers (frontier_explore/DetectFrontiers)
  > Self-defined computational service that detects the frontiers from map. See section 2 for details.

1.5 Parameters

~ `minclasssize` (`int` , default: `135`)

> A frontier class is valid only if its size is larger than this `minclasssize` . The optimal value of this parameter is highly related with the resolution of current OccupancyGrid map. 135 is a recommended value for the map with resolution 0.01.

~ `minexploredist` (`double` , default: `1.3`)

> The minimum distance of each exploration action. A frontier target is valid only if the distance between the target and the robot is larger than this `minexploredist` . 1.3m is a recommended value for the simulation environment.

1.6 Required tf Transfroms

`map -> base_link`

> This transformation is typically provided by SLAM nodes, e.g. *slam_gmapping* node.

2. detect_frontier_server

The detect_frontier_server node provide a computational service called *DetectFrontiers*. It takes in the OccupancyGrid map and outputs a queue of qualified frontier centroids that are detected from the map. The computation part is implemented in the *OccupancyMap* class.

2.1 Services provided

- detect_frontiers (frontier_explore/DetectFrontiers)

Definition of *DetectFrontiers.srv*

```
nav_msgs/OccupancyGrid raw_map
int32 minclasssize
~~~
geometry_msgs/Point[] frontiers
```

Request message: `raw_map` is the OccupancyGrid map obtained from laser scan, `minclasssize` is the parameter set in frontier_explore_node.

Response message: `frontiers` is a vector of geometry points, which represents the geometry centroids of the frontier classes detected from map.