Assignment 5: Project

Version 2017-06-04

Intent: To analyse a set of specifications, design, test, document and practically evaluate code to perform analysis of sensor data and simple actions based on data on a simulated robotic platform.

Individual Task

Weight: 40%

Task: Write a series of components using the ROS CBSE framework that will process data originating from a range of sensors on a simulated robot. Employ suitable number of threads per component and data structures that enables time synchronisation and subsequently interrogation of data to allow simple actions of a robotic platform. Supply appropriate auto-generated documentation utilising inline source mark-up. Exploit unit testing framework with test cases evaluating code.

Students can select from two projects

1) Probabilistic Road Maps
2) Exploration of Frontiers and Path Execution

Rationale: In a Mechatronics System, sensors produce data at varying rates. Decisions need to be made based on correctly associated data in near real-time. Threading and synchronisation are ways to ensure the system performs as intended, with guarantees on the responsiveness of the system to incoming data changes, processing constraints and system behaviour. Functions that exploit the data require unit testing to ensure they behave correctly. Documentation of your own code allows other developers to utilise it as intended and anticipate outcomes, in the same fashion you use a number of APIs (ROS/OpenCV).

Due: Sunday 25th June 23:59

Occupancy Grid Map (OgMap) Specifics

An OGMap is a discrete representation of the environment, used to denote space known to a robot. In its most basic implementation the value of each cell in the map represents the robot's knowledge of the occupancy of the space. That is, either occupied, free-space or unknown. The map resolution is specified by user, resolution is valid if it is an integer (whole number), based on the formula: pixels = map_size / resolution.

The framework for receiving data and publishing the OgMap (encoded as an OpenCV image) is provided as part of the tutorial sessions in Week 12 (local_map and map_to_image_node). The map size is known (20x20 meters), each cell is 0.1m the robot can be moved about using rqt.

Both projects require querying the Occupancy Grid Map (OGMap). Any transformation between local (ogmap map centre) to global (robot position) and image (pixel) coordinate system should be unit-tested. In the implementation of OgMap supplied, the robot position coincides with the centre of the OgMap.

Project 1: Probabilistic Road Map (PRM)

The probabilistic roadmap planner is a motion-planning algorithm in robotics, which solves the problem of determining a path between a starting and a goal configuration of the robot while avoiding collisions.

The basic idea behind PRM is to take random samples from the configuration space of the robot, testing them for whether they are in the free space, and attempt to connect these configurations to other nearby configurations.

The probabilistic roadmap planner consists of two phases: a construction phase, and a query phase. In the construction phase, a roadmap (graph) is built, approximating the motions that can be made in the environment. First, a random configuration is created. Then, it is connected to some neighbours, typically either the k nearest neighbours or all neighbours less than some predetermined distance. Configurations and connections are added to the graph until the roadmap is dense enough. In the query phase, the start and goal configurations are connected to the graph, and the path is obtained by a shortest path query algorithm.

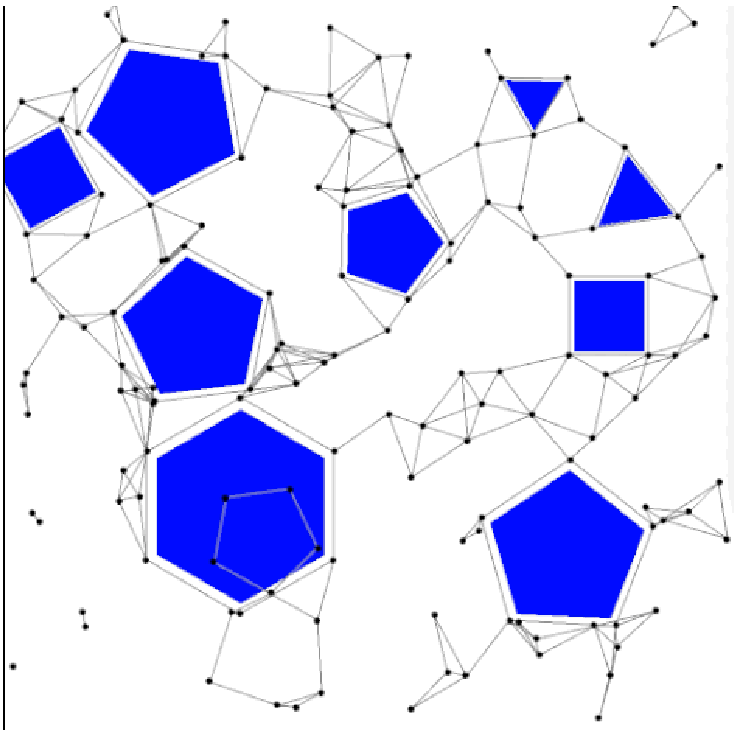

Figure 1 - `"Motion in Environments with Similar Obstacles" Robotics: Science and Systems. 2009 (Eric Scot, Wikipedia, last viewed 4th June 2017)

Specifics

Configuration space can be obtained from the OgMap taking into account the diameter of the Robot (20cm). The path needs to be published to **/path** topic using the **PoseArray** message (http://docs.ros.org/api/geometry_msgs/html/msg/PoseArray.html).

Requests for the path arrive via the **RequestGoal Service Call**. The pose of the robot is published on the **/odom** topic while the OgMap **/map_image/full**. Either publish or allow the component to produce an image that shows all the configurations (nodes) of your graph as circles and all connections as lines.

For P/C Grade:

The starting configurations is the current position of the robot, and the goal position is specified via the RequestGoal Service Call. Create a Probabilistic Road Map (limit connectivity to 5 nearest neighbours), terminate the building phase when start and end goal are connected via PRM. Query the graph for a **shortest path** connection, any method is acceptable.

For D/HD grade:

Enable reuse of the PRM in the next query phase by adding more configurations using the current OgMap.

Bonus Mark (additional 15% of marks)

Narrow passages present significant challenges for PRM implementations. When creating the PRM and sampling configurations address this issue. NO additional guidance will be provided for Bonus Mark problem.

Project 2: Frontier Based Exploration and Robot Control

Exploration is the act of moving through an environment, thus allowing building a map of the environment which can be used for subsequent navigation. A good exploration strategy is one that generates a complete or nearly complete map in a reasonable amount of time.

One approach for exploration is based on the detection of frontiers, regions on the border between known free space and unknown space. From any frontier, the robot could see into unexplored space and add the new observations to its map. From each new vantage point, the robot may see new frontiers lying at the edge of its perception. By pursuing each frontier the robot can build a map of every reachable location in the environment. OgMaps can be leveraged as spatial representation suitable for determining frontiers. For an explanation of frontiers in OgMaps revert to http://robotfrontier.com/frontier/simple.html

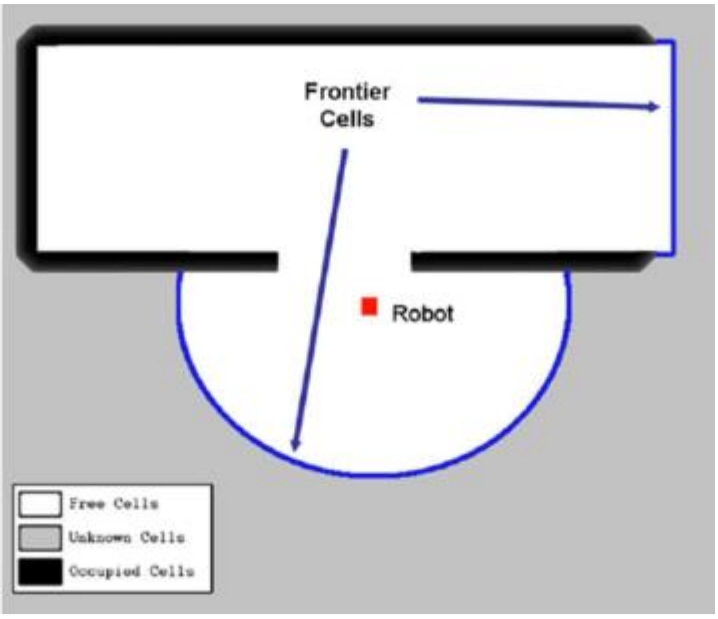

Figure 2 - Example of frontier, from Juliá, M., Reinoso, O., Gil, A., Ballesta, M., & Payá, L. (2010). A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence, 23(4), 473-486.

If a path to a frontier is supplied, with a series of waypoints, a control strategy needs to be put in place to execute the path (closed loop control strategy). This typically means a controller over the velocity and the turn rate of the robotic platform. In the simplest implementation, Iteratively going through the waypoints, aligning the robot via turn rate to the desired angle connecting the two points, zeroing the turn rate and then using a pure pursuit controller with a set velocity (refer to the link below)

http://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf

Continuously checking the waypoint is reached as required. Once reached, repeat the process untill the last waypoint is reached. On the last waypoint ensure the angle of the robot matches the specified angle of the GoalPose.

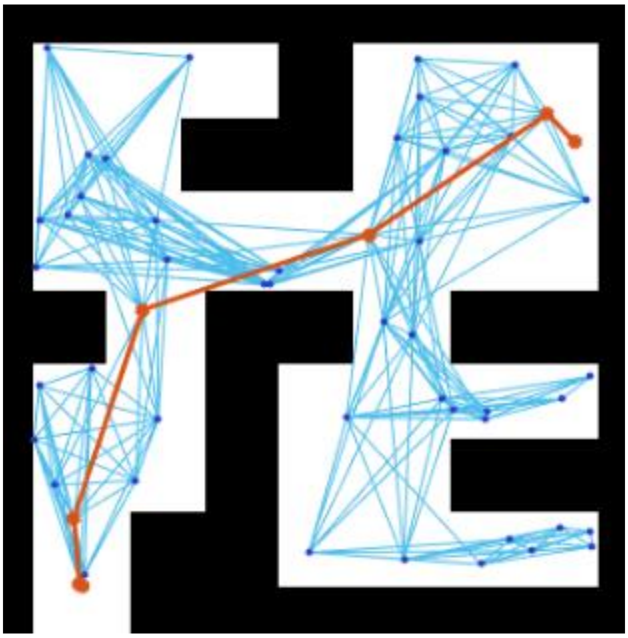

Figure 3 - Example of a path (as series of waypoints) between start/goal as a series of nodes the robot needs to traverse through

Specifics

Frontier cells can be obtained from the OgMap. Requests for the path need to be made via the **RequestGoal Service Call**. The pose of the robot is published on the **/odom** topic, while the OgMap is published on the **/map_image/full** topic. The component should either publish as an image or directly display an image that shows all frontiers and waypoints as points with connections as lines.

The path will be supplied via **/path** topic using the **PoseArray** message (http://docs.ros.org/api/geometry_msgs/html/msg/PoseArray.html).

For P/C Grade:

Compute frontiers every 10s, compute a GoalPose from the frontiers as the closest point to the current robot position (shortest distance). Move the GoalPose into a valid pose (being in free space). Ensure that the angle of the robot pose (heading – yaw) is such that it points into unknown space.

For D/HD grade:

Implement the waypoint following code for the robot as a combination of on the spot turns and pure pursuit.

Bonus Mark (additional 15% of marks)

In order to make the waypoint following smooth, devise a waypoint following strategy that results in smoother paths. NO additional guidance will be provided for Bonus Mark questions.