EECE 5550 Mobile Robotics Final Project

David M. Rosen

Due: Dec 17, 2021

Motivation and background

Over the previous two decades, mobile robots have begun to play an increasingly important role in *disaster response*, providing emergency response teams with both improved situational awareness and physical control in hazardous environments.[1] One particularly prominent application is *reconnaissance*: in this task, emergency responders will deploy mobile robots to quickly survey a potentially dangerous environment (such as the interior of a collapsed building) to identify potential hazards and/or the locations of victims. This information can subsequently be used to develop an effective action plan that minimizes responders' exposure to risk.

Objective and problem formulation

In this project, you will design and implement a complete autonomous system to perform reconnaissance in a simulated disaster environment. Specifically, when introduced into a *closed* but *initially unknown* environment, your system should do the following:

1. Generate a ***complete*** map of the environment,
2. Report the locations of any victims present in the environment.

For the purposes of this exercise, we will use AprilTags as stand-ins for simulated victims, and adopt occupancy grid maps as our environmental representation. Therefore, the specific outputs of your system corresponding to points (1) and (2) above should be:

1. A ***complete*** occupancy grid map of the environment.
2. A list of AprilTags discovered in the environment. Each element of this list must specify:
   - The ID (number) of the detected tag
   - The pose of the tag, *expressed with respect to the map frame.*

Rules of the game

You may construct your system using any off-the-shelf components that you like. For example, you may want to make use of the Cartographer package that we saw in Lab 3 in order to build occupancy grid maps, the move_base package from the ROS navigation stack to command your robot to move around the environment, and/or a frontier exploration package such as explore_lite.

[1] See (for example) Chapter 60 of the *Springer Handbook of Robotics* for a recent survey on the use of mobile robotics in disaster response.

However, you should consider that a baseline system constructed *solely* using off-the-shelf components may not perform very well on the target task. For example:

- While a basic frontier-based exploration planner (such as `explore_lite`) can ensure a complete *occupancy grid map*, it does not address the goal of searching for "victims" (AprilTags). You might therefore consider either modifying `explore_lite`, or implementing your own planner, in order to improve your system's ability to find all of the tags present in an environment.
- The (estimated) pose associated with an AprilTag detection can be rather noisy, especially when the tag is viewed from a distance. You might want to think about possible approaches for ensuring that the pose of each detected tag is estimated accurately in the final map.

In addition to off-the-shelf ROS packages, when implementing your system you may find the OpenCV library (especially the image processing module) useful for operating on occupancy grid maps, and/or the GTSAM library useful for performing parameter estimation, should your approach require it.

Deliverables

Your team should submit the following:

1. The code for any ROS nodes that your team implements or modifies. This code should be committed to your Github repository.
2. The launch file(s) used to start up your system.
3. A report describing the design of your system. This need not be lengthy, but it must contain the following elements:
   - A clear description of your system's construction. Operationally, this means that a skilled practitioner should be able to reproduce (that is, *reimplement*) your system *without* looking at your team's code. Practically, this will primarily involve detailing the interfaces and algorithm pseudocode for any custom or modified ROS nodes that your team implements, together with brief descriptions of any other off-the-shelf components that your system employs.
   - The rationale for your design: that is, *why* your team adopted the design that it did, and what makes that design a good solution for the task at hand.
   - Experimental results from an example application of your system. This should include a (brief) description of the environment in which your system was tested (i.e. type of location, approximate size of environment, number of tags present), and a visualization of the final occupancy grid map and detected AprilTags reported by your system.
4. Recordings of:
   - The RViz visualization for the experimental example submitted in your report
   - The robot itself as it was running in the same experimental scenario.

Evaluation

Your submission will be evaluated on the following criteria:

1. **Soundness of design:** That is: is your proposed design well-suited to the task? For example: Can you show that your design is (eventually) *guaranteed* to locate all tags that may be visible in a given environment? Is your system's exploration strategy efficient with respect to time/speed?

2. **Empirical performance:** Does your system work well in practice? To provide a common baseline for comparison, we will make available a (large) test environment in ISEC 138 on **Thursday Dec 16** for each team to demonstrate their robot's performance.