CMPUT 412 Competition 5: Creative Project (Photobooth Robot)

Jae-yeon (Leo) Yoon
Sumeyye Aydin

Objectives

Generally, the objective of this competition was to combine gained knowledge throughout the course into one applicable program. The concepts include:

- State Machines (smach)
- Robot Movement
- Odometry
- Mapping and Localization via Simultaneous Localization and Mapping (SLAM)
- Map construction via GMapping
- Camera Calibration
- Feature Detection via ORB
- AR tag Detection
- Navigation via Adaptive Monte Carlo Localization
- Practical Project pitch skills

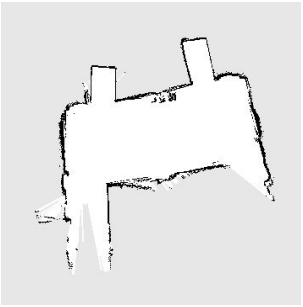
Mapping concepts were applied - with SLAM - to construct a map of a designated area on the second floor of the Computing Science Center at the University of Alberta. Through construction of a relatively accurate map, our robot was able to navigate to a desired waypoint where our 'client' would be seated.

Background

This competition was inspired by applying practical project pitch strategies and achieving set milestones to demonstrate a proof of concept. The Robot Operating System (ROS) was the main software component used in our Turtlebot 2 machines. Documentation of ROS can be found at http://wiki.ros.org/ .

For global localization of the area, the robot used both the map.pgm and map.yaml files to simulate its relative location in the real world environment. As stated in the objective, the .pgm and .yaml files were produced through SLAM via GMapping. Documentation on GMapping concepts can be found at http://wiki.ros.org/gmapping .

In general, the GMapping algorithm distributes a Rao-Blackwellized particle filter in the simulated environment to predict both the location of the robot and the position of obstacles. Through information received from our visual sensor (Asus Xiton Pro Live Camera), the algorithm alters its previous predictions to generate a more accurate location in real-time.



Generated map of designated area

During the navigation portion, Adaptive Monte Carlo Localization (AMCL) - a probabilistic localization algorithm - was used to keep track of the robot's location within the generated map along with the real world environment.

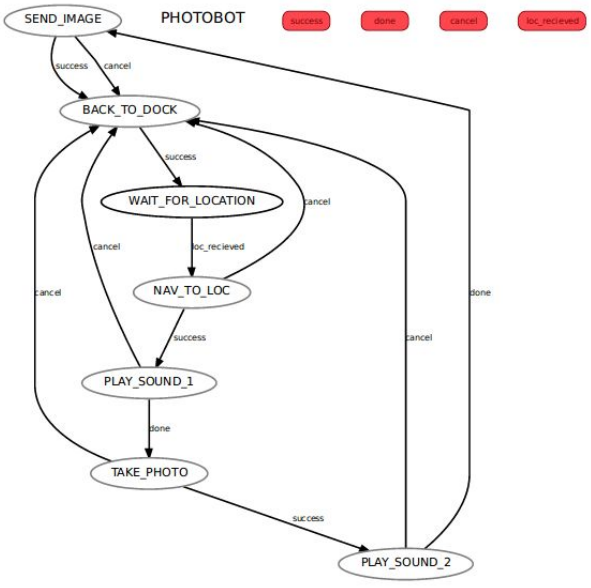
Hypothesis

Our competition algorithm allows the Turtlebot 2 to navigate the given environment to the desired location, take a photo of the client, and return back to docking to maintain its battery charge.

Materials

- Turtlebot 2 Robot
- Asus Xtion Pro
- ROS
- Logitech USB Camera

Method



State Machine Concept

Localization

After undocking from the docking station (via basic movements based on odometry), we run the *global_localization()* function provided by rospy to localize the robot within the environment - regardless of starting location. The *move_base/clear_costmaps* service is called to allow an external user to message *move_base* to clear obstacles in the costmaps generated by *move_base*.

Navigation

The robot was designed to navigate to a single waypoint within the competition zone. The waypoint was estimated through trial and error within rviz. The task of navigating the environment and avoiding potential obstacles was left to ROS's package, *turtlebot_navigation*.

Face Detection

By using haar files (provided by https://github.com/Itseez/opencv/tree/master/data/haarcascades) and OpenCV packages, the algorithm was able to reliably detect faces and capture their photographs.

Docking

The auto-docking feature was taken care of by kobuki's proprietary auto-docking package (more information can be found here http://wiki.ros.org/kobuki/Tutorials/Automatic%20Docking). After setting a designated waypoint near the docking station, the package locates the station and proceeds to dock the robot automatically.

Results

Fortunately, our robot was able to perform the tasks set out by our self-set goals, including: map building AMCL (non-automated), collision free navigation of environment, auto-docking, and image capture of faces.

The video of our photobooth robot's proof of concept can be found here:
https://www.youtube.com/watch?v=z3kAj6gYjwY

Discussion

The most difficult and time consuming issue to tackle during the competition was integrating all the parts into one cohesive program. By solving smaller problems and then integrating them into the final product, we remained organized. Even though our robot is functional in terms of individual portions, many use cases must be covered to make this project remotely viable in a real world environment. We were unable to build an user interface due to time constraints.

Conclusions

In the end, our goal was to further understand SLAM, feature detection, navigation of a map via AMCL, and practical project proposal strategies. From the results of the competition, we can see that there is definite room for improvement, as our robot would not be able to function dynamically in the real world, this demonstration proved to be just a proof of concept, and nothing more. Our robot was able to satisfy the goals we set out to accomplish during this project. Overall the results of our program supports our hypothesis, due to the fact that we were able to meet all of our project milestones. This exercise was invaluable as it taught us real world situations where we as developers would have to pitch our ideas to investors.

Further improvements would include:

- A functional User Interface
- Dynamic mapping
- Dynamic waypoint setting/User input waypoint setting
- Better mapping strategies