COMP-150DR Final Project Report:
CARoL: Coordinated, Automated Robots of Levity

Azmina Karukappadath, Yuelin Liu, Sam Weiss
Tufts University, COMP-150DR Developmental Robotics

Abstract. With the holiday season fast approaching, our group decided to focus our project around creating a caroling multi-robot system. It is an interesting topic because it not only embraces the holiday spirit, it can also be broken into sub-problems that are relevant to developmental robotics, namely: 1) door detection, 2) coordination in task completion, 3) learning rejection, and 4) developing ”motivation.” This final paper will outline the steps we took to develop CARoL, and detail the concepts we were able to accomplish as well as those we hope to work towards in future work.

1 Introduction

Developmental robotics is an interdisciplinary subject that builds upon the idea that, in order to create autonomous agents that are intelligent and adaptive, they should go through a developmental period like humans do. One key concept for development robotics is the verification principle, which emphasizes that agents cannot learn, and subsequently adaptively apply, something it cannot verify.[1] By making robots capable of modifying their behavior according to their experience of interacting with the environment they are situated in, they are more equipped to function in the real world, where everything in the environment is unpredictable.

For the course final project we incorporated this idea and created a system that could learn and adaptively modify its behavior in achieving some tasks. Since the start of the holiday season coincides with the end of the semester, we thought of the idea of creating a multi-robot system that roams the second floor of Halligan and carols at each office. We wanted explore the problem of multi-robot coordination as collaboration is essential in solving real world problems of a larger scale: we could solve a problem faster if we could let a team of agents coordinate among themselves to properly divide up the problem into smaller parts, have a them each be responsible for tackling a part.

To better describe what this project envisions, we divide the problem into 4 subproblems:

1. Door Detection,
2. Coordination in Task Completion,
3. Learning Rejection, and
4. Developing Motivation

This paper will elaborate on the details concerning our project in terms of those subproblems, and discuss our approach to solving them.

2 Problem Formulation

2.1 Door Detection

Our project builds upon the assumption that the robots have already obtained the floor plan of the hallways of the second floor of Halligan; however, since they have to carol in front of doors, they would have to identify the locations of different office doors along the walls. Specifically, the door detection problem asks, when given a floor plan of the hallways, how a robot would learn the positions of the doors so that they know where to stop and carol.

2.2 Coordination in Task Completion

Our problem formulation for the Coordination in Task Completion tasks is as follows: Given a set of doors to visit, visit all doors and carol at them in an efficient way. This means no visiting a door once it has been caroled at, as well as prioritizing doors that are most likely to have people interested in hearing carols.

2.3 Learning Rejection

If there is nobody in the room, or if the person in the room does not want to be caroled at, our robots should be able to interpret ”rejection,” and modify its caroling behavior.

2.4 Developing Motivation

In order to develop motivation in our robots, we need to find some way to quantify how successful they are at caroling. Once we have developed this metric then motivation is simply a process of maximizing that success criteria.

3 Primary Results

3.1 Navigation

Our first task in developing a caroling robot was to develop a way to efficiently record and then later navigate to all of the doors in Halligan Hall. Prior to the start of our project, a full map of the second floor of Halligan had been recorded, so this was not a technical challenge that we had to address. However, this map contained only laser distance readings that recorded the positions of the walls, not more advanced information like the location of doors. Additionally, the office in Halligan are located on the long, relatively featureless hallways. This had two implications for us: doors were impossible to locate through the map along, and we weren't able to navigate to them with high accuracy. We weren't able to determine the location of doors solely by analyzing the map because it came with a fair amount of noise. Some features like columns that extended 6 inches from the wall didn't seem to appear in the map at all, while some features in the map appeared to have no real world counterparts. While we were not able to determine the location of doors in the map, we were able to do so in the real world. During our first two working sessions we drove the CARoL robot around Halligan to every relevant door, and simply recorded the position of the robot in the map. This allowed us to determine a location and a direction for each door. We recorded this information in a CSV file for later use. We further augmented this list of room numbers and coordinates with a list of names corresponding to the owners of specific offices. Some considerations were made for rooms which contained multiple offices, we decided to label these rooms as containing ”friends.” With our list of destination collected we were now ready to begin actually navigating to them.

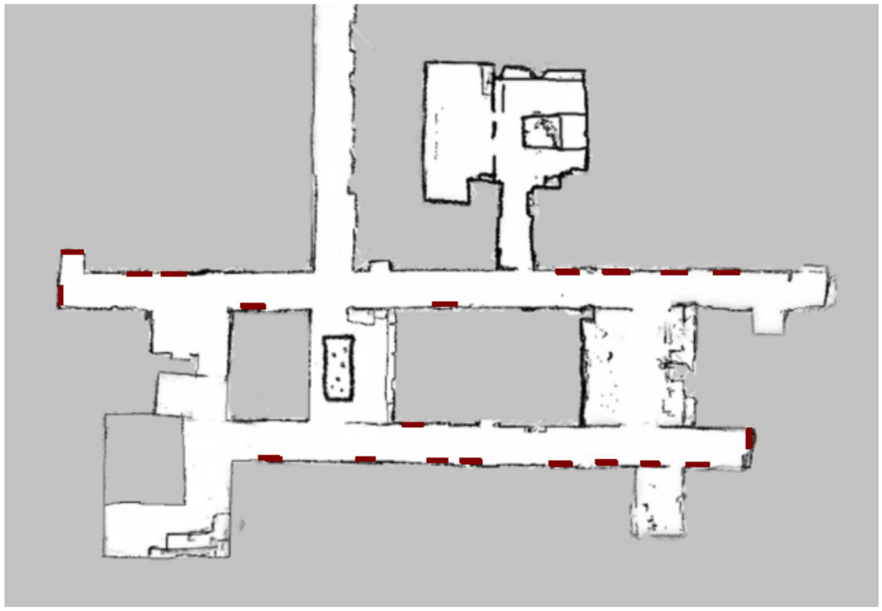

Fig 1 Navigational Map of Halligan - Doors Marked in Red

During our initial testing of our newly gathered coordinates we encountered some significant problems that plagued the following stages of this project. As mentioned earlier, many of the doors we were attempting to navigate to are situated in featureless corridors. These featureless corridors are navigational nightmares for robots and humans alike. We as humans are able to pick up on visual cues like door frames, posters, signs, and more. But our CARoL robot is limited to only laser scan and point cloud data at around knee height. Because of this poverty of useful information, our robot was unable to accurately determine its location in the hallways, making it impossible to precisely navigate to the desired position and orientation. Additional confounding variables like students walking in front of the robot, trash cans that seem to migrate up and down the hallway, and inherent inaccuracies in the odometry data coming from the wheels meant that the robot was at best within 2 feet and 60 degrees of the desired position and orientation. While this may sound reasonable in a building with thousands of square feet of floor space, it was unfortunately not accurate enough for our purposes. The AMCL navigation software used by our turtlebot is a robust Monte Carlo localization algorithm.[2] Because this localization algorithm only gives an approximation based on a Bayesian inference, it’s impossible for it to navigate to exact points in space. Unfortunately this was not a problem that we had time to adequately address during the course of our project.

Our proposed solution is a fusion of two different existing techniques. We propose a solution that uses the efficient and largely accurate Monte Carlo localization algorithm[2] for building level navigation. Because our existing solution was able to navigate through a variety of challenging situations and environments we see no need to abandon this solution entirely. The AMCL navigation system we used was effective in hallways and crowded areas, but we propose adding an additional navigational step at the end of our conventional navigation. The goal of our navigation is to line us up with the target doorway so that we are able to detect whether it is open and if we should carol. This is simply not possible when our robot is facing the wall two feet to the right of our destination instead of the door itself. In order to address these problems we posit that a visual system to detect doors and their location relative to the robot would allow the robot to navigate to a proper location more accurately. Presented below is a high level flow chart indicating how our proposed system would navigate to a door:

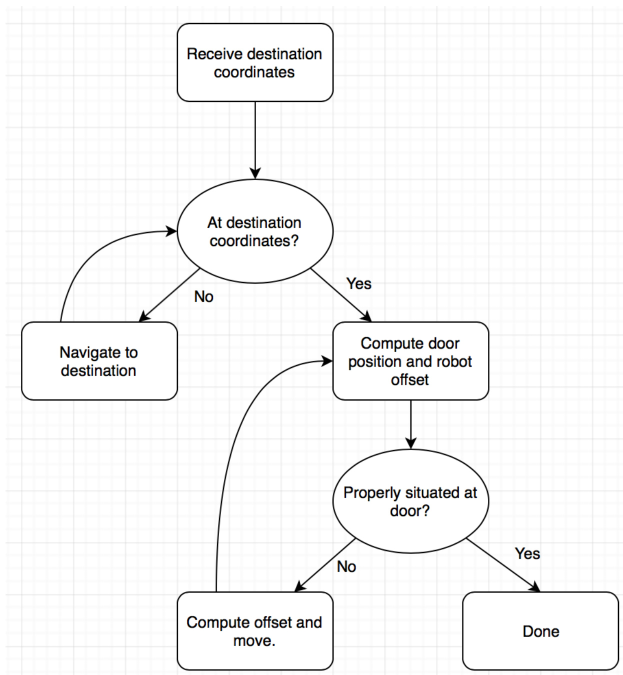

Fig 2 Dual Navigation Flow

This system would simplify the mapping process and make navigation more accurate and useful. Algorithms like those proposed by Fernndez-Carams et. al present a robust system to determine the location of doors in a scene.[3] Such a system would also enable the CARoL robot to autonomously determine the locations of doors without the time consuming manual effort that we spent driving the robot around Halligan and attempting to determine the optimal orientation and positioning. To conclude, our navigation systems were effective to a degree, but further refinements are required to make a truly robust and effective caroling robot.

3.2 Door Detection

Once our robot has navigated to a door, it must use its laser scanner to determine if the door is open. An open door means that there is likely someone inside and that CARoL should perform a song and dance. In our project proposal we hoped to implement a machine learning algorithm to detect whether or not a given door was open, but due to some technical limitations of the laser scanning system this proved infeasible. We decided to implement a more conventional algorithm (outlined below) that was effective when positioned correctly. Our algorithm is designed to be tolerant to one of the major weaknesses of the hardware we used, namely that it produced invalid readings for distances that are too short or too long. We found that our laser scanner would yield a `Not a Number` value if the distance it was trying to detect was above 10 meters or below 0.5. This can lead to two different situations where the same laser readings can mean a door is open or closed which we've outlined below.

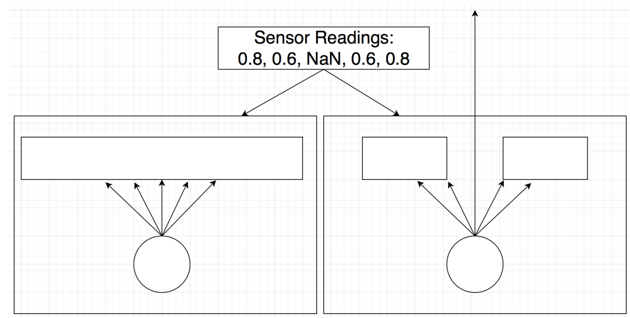

Fig 3 Potential Sensor Confusion

In order to try and account for these irregularities in sensor readings we implemented a best effort algorithm outlined here:

1. Wait for laser sensor reading.
2. Subsample the middle 1/2 of the laser readings.
3. Iterate over readings:
   (a) If a laser reading is NaN, discard it.
   (b) Count laser readings below threshold.
   (c) Count laser readings above threshold.
4. If there are more laser readings above the threshold than there are below the threshold, the door is open.

This algorithm proved to be sufficiently accurate for our purposes. We tested it on a six open and closed doors in Halligan with perfect accuracy when positioned correctly. However, as mentioned in the previous section, this correct position was the exception not the norm when the robot was operating under normal conditions. In normal operation the robot was only oriented and positioned in such a way to detect a door around 30-50% of caroling attempts. To conclude, our door detection algorithm was a best effort to extract meaningful information from an inconsistent sensor that was ultimately marred by inaccurate position.

3.3 Attention Acquisition

Having now developed a way to navigate to potential caroling targets and detect whether their door was open, we decided to tackle how to acquire a user's attention if their door is closed. We broke this down into a fairly simple process with some unexpected complexities. We decided to try and "knock" on doors in the way that human carolers do. However, since our robot lacked any appendages to knock with we were forced to resort to other auditory methods of getting a carolee's attention. In order to accomplish this we turned to psychology and the Cocktail Party Effect.[4] The Cocktail Party Effect describes two main phenomena, humans' ability to focus in on a specific auditory stimulus in a noisy environment, and our ability to pick out important information in noise we're not focusing on directly. The most famous example of the latter phenomena is how attune we are to our own names. It's very easy for us to notice when our names are mentioned even in sound we're not focusing on directly. For this reason we decided to have our robot speak the name of the person whose room it was caroling at. This proved to be quite successful with many professors opening their doors despite the somewhat muffled speakers that the robot was equipped with. Our attention acquisition algorithm is as follows:

1. Navigate to room and detect that door is closed
2. Loop 3 times
    (a) Speak "Hello, name please open the door"
    (b) Ten times per second detect if the door is open or closed
    (c) Wait 10 seconds
3. If the door opened, then carol
4. If the door did not open, record the failure and move to the next room.

This algorithm makes a reasonable effort to get the attention of whoever is in the room without being overly annoying. It also detects that the door is open as soon as it opens so that caroling can begin immediately. When the robot is aligned correctly with the door it was effective in our tests.

3.4 Learning Rejection

An essential component is experiential based learning which we implemented in two parts, one designed to learn within a caroling session and other other designed to allow the robot to learn across caroling sessions. Both of our machine learning algorithms work on a priority system that we've developed whereby rooms with higher priorities are more likely to be caroled successfully. The robots maintain a separate session and global priority for each room that it can potentially carol at, using the session priority to determine which room to carol at next. These priorities are then updated by on online, experiential machine learning system.[5] Initially, the priorities of all rooms are set to 100, indicating that we have no evidence to indicate how likely they are to open the door to carolers. At the start of a caroling session, the CARoL robot loads the existing priorities and selects the room with the highest priority. It then navigates to that room and attempts to carol. If it is unable to carol successfully it updates the session priority to be 1/2 the value it was before the attempt. If it is successful in its caroling attempt it sets the session priority of the room to 0 indicating that it no longer needs to carol at that room, while doubling the global priority. The robot continues to operate in this way until all rooms have a priority below some threshold. If a room was never caroled successfully during a caroling session then its global priority is halved. This system means that the robot visits rooms that are likely to be successful caroling sessions before less cheery rooms, and that it tries to carol that more times. Additionally, rooms are not visited again if the CARoL bot carols at them successfully. This proved to be effective in our testing with the robot learning to avoid some rooms that were always empty/hostile.

While this system was effective for us, we investigated more advanced methods of machine learning and success definition. One possible improvement that we think could have a material improvement on our project would be a more refined success detection system. In the course of our research we came across a paper by Matsugu et. al describing a system of neural network based facial expression recognition.[6] We experienced some situations where a door was opened for our robot, but where the person who opened the door either closed it again quickly or was actively hostile towards the robot. Since the door opened in these situations CARoL counted them as successfully caroling at the doors and increased the priorities of those rooms. In the future, we would like to integrate a facial expression detection algorithm to detect whether the people who opened the door were happy and interested or if they were angry and disinterested by using feature detection to detect between the seven universal emotions,[7] which have features that are different and simple enough to be accurately distinguished between. We would then update priorities based not only on whether or not the door was opened, but also what the user's reaction was to our caroling robot.

4 Future Work

4.1 Multi-Robot System

Due to time constraint, we were unable to fully implement multi-robot coordination; nonetheless, we have examined different types of multi-robot systems, and considered their advantages and disadvantages, respectively. Depending on the specifics of the problem, one system might be a more reasonable design for the conditions than another.

A multi-robot system involves multiple mobile agents, and in cases where the overall task requires the robots to travel a wide range of locations, it may lead to more efficient solutions by breaking the problem down into smaller units which were then assigned to individual robots to tackle in parallel. In their review article, Yan et al. discussed the differnet variables involved in multi-robot systems.[8] We will discuss each aspect with regard to CARoL.

First of all, individual robots involved in a multi-robot system can either have the same capabilities as their teammates, or they can each specialize in one or several skills. The former qualifies as a homogeneous system, and the second qualifies as a heterogeneous system.[8] For this project, all available Turtlebots are identical in terms of affordance and set up; therefore, it would have been a homogeneous system. For each round, each robot would take up the task of caroling in front of one of the office doors. With 4 robots running in parallel, we would be able to finish the overall task sooner. We consider this design appropriate, since each subtask would be largely independent from others. For another project, it might be reasonable to involve a team of heterogeneous robots that have different affordances and skills. A heterogeneous team would be advantageous in situations where the (sub)task at hand is too complex or costly to have one robot to complete on its own. For example, if we added an door-opening aspect to this project, and it is too difficult to attach a robot arm onto the Turtlebot, we could involve a robot specialized in opening doors, constructing a heterogeneous team. Having a heterogeneous team, however, would require a higher degree of coordination, as the task assigned to one robot is dependent on the status of that of another.

A multi-robot system could also be either cooperative or competitive.[9] In a cooperative system, the agents in the system each plays a part in contributing to the overall task, with the goal being maximizing the overall score. On the contrary, in a competitive system, the goal for each individual robot would be to maximize its own score gain. For our project, there is no need for the robots to be competitive; nonetheless, an interesting project would be to implement a competitive race between two cooperative systems, where success is measured by the total number of rooms each system has visited. Each team will have to both coordinate internally and strategize against the other team in order to win.

Although both involve planning to some degree, coordination in a multi-robot system could be either static or dynamic.[8] A static system bears resemblance to classical planning, as it involves computing well-defined steps to accomplish a task before we dispatch the robots. This is not very useful to our project, as our robots not only have to actively adapt to the dynamic environment of the building, but also have to be able to devise new strategies based on whether the rooms are occupied in reality. Dynamic coordination requires the robots to be constantly communicating, updating the global state of the system, and reactively compute the actions each robot should perform.

To create this multi-robot system, we could implement a centralized or decentralized architecture.[10] In a centralized unit, a central machine would have the information about the state of all the doors in Halligan, including the caroling histories of each of the robots. This central machine would then create the tasks for the individual robots, and update the information on Halligan whenever needed. However, this system would not be very robust, since the central machine failing would cause the entire system to fail. A decentralized architecture, would not have this drawback since each robot or cluster of robots is autonomous in their decision-making process. This would cause the system to be more robust, but introduces increased difficulty in communication between robots - this reduces the number of optimizations that can be performed to make the caroling behavior as efficient as possible.[10]

5 Conclusion

As we can see, there are many interesting variants of this project for people to implement and realize. Over the course of this project, we learned to design and implement a robot to perform a certain task. Through future work, we hope to implement better methods of judging caroling attempts as successful and introducing a coordinated multi-robot system to increase efficiency. We hope to be able to work with Turtlebot again in the future, implementing more projects that will bring joy to the Computer Science community in Halligan.

References

1 A. Stoytchev, "Some basic principles of developmental robotics," *IEEE Trans. on Auton. Ment. Dev.* **1**, 122–130 (2009).

2 S. Thrun, D. Fox, W. Burgard, *et al.*, "Robust monte carlo localization for mobile robots," *Artificial Intelligence* **128**(1), 99 – 141 (2001).

3 C. Fernández-Caramés, V. Moreno, B. Curto, *et al.*, "A real-time door detection system for domestic robotic navigation," *Journal of Intelligent & Robotic Systems* **76**, 119–136 (2014).

4 B. Arons, "A review of the cocktail party effect," *Journal of the American Voice I/O Society* **12**(7), 35–50 (1992).

5 T. M. .-T. M. Mitchell, *An Overview of Machine Learning*, McGraw-Hill, New York :.

6 M. Matsugu, K. Mori, Y. Mitari, *et al.*, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks* **16**(5), 555 – 559 (2003). Advances in Neural Networks Research: IJCNN '03.

7 P. Ekman and D. Keltner, "Universal facial expressions of emotion," *California mental health research digest* **8**(4), 151–158 (1970).

8 Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems* **10**(12), 399 (2013).

9 S. J. S. J. Russell, *Artificial intelligence : a modern approach /*, Prentice Hall, Upper Saddle River, N.J. :, 3rd ed. ed. (2002).

10 C. M. Clark, S. M. Rock, and J.-C. Latombe, "Motion planning for multiple mobile robots using dynamic networks," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, **3**, 4222–4227, IEEE (2003).