CRA-W Computing Research Association Women

the coalition to diversify computing

CREU 2016-2017 Final Report: Cooperative Robotics for Target Search and Coverage

CREU Team Mentor: Dr. Sambit Bhattacharya, Fayetteville State University
Team member: Daphne Faircloth, Fayetteville State University
Team member: Catherine Spooner, Fayetteville State University
Team member: Brittany Jennette, Fayetteville State University
Team member: Donny Lopera, Fayetteville State University
Team member: Ashlee Gaskins, Fayetteville State University

I) Goals and Purpose

When one thinks of robotics, often the images that come to mind are the automations introduced into our industries or the usually frightening villains of science fiction films. Even for most scientists, it is still a relatively new field of study and application.

One such application is being pursued by the National Aeronautics and Space Administration (NASA) and has led to a collaboration with the Moses Biological Computation Lab at the University of New Mexico (BCLab-UNM). The result of this collaboration is the NASA Swarmathon, a search-collect-return algorithm writing competition for undergraduate students at Minority Serving Universities and Colleges.

Swarm robotics involves the deployment of multiple simple robots to perform a variety of tasks rather than utilizing a single agent. As citizens of Earth, we have seen the considerable investment of time and resources the Curiosity Project [1] required. Imagine, then, the Mars rover experiencing a mechanical issue that renders it inert. Nearly 3 billion dollars has just been reduced to a useless pile of electronics. Imagine instead that the same price tag yielded not a single complex agent, but a hundred smaller, simpler agents. The loss of one would not be a catastrophic event, and even a 50% loss would still leave fifty rovers on Mars, exploring, recording, transmitting, and collecting samples.

This research group joined this competition with several goals in mind: improve coding and research skills, encourage future participation in research and technology, and hopefully gain new search algorithms for use by NASA in swarm robotic exploration. We also intended, in our Outreach endeavor, to encourage the members of our high school team to explore future research opportunities and potential careers in Science, Technology, Engineering and Math fields

II) Related Work

a) Swarmathon repository

The GitHub Swarmathon repository [2] created by BCLab-UNM provided the base code for the competition. Our first task was to read this repository and gain an understanding of what the variables represented, what each function did, and how the rover processed and shared data.

For most of the team, this was also an introduction to C++, which posed its own challenges and rewards.

b) OpenCV and the AprilTag Library

Before our team even officially entered the competition, we were working on ways to improve target detection and collection. Swarmathon organizers announced that cubes with identical AprilTags printed on all sides would be collected in the next competition instead of flat AprilTags being digitally returned.

Using the OpenCV library [3] and a demo developed at MIT available from the AprilTags C++ Library [4], the team worked on creating a training bank of positive and negative annotations as a machine learning exercise. We used a training bank of 834 images for our detector and gathered additional images to use as a test bank. Of the 255 cubes in the test images, our detector correctly identified 101 while the MIT demo found 181. However, we also had 75 reported as false positives and MIT had none as illustrated in Fig. 1.

We determined we needed to increase our training annotations to at least 1000 annotations and perhaps as many as 3000. However, it was during this time that Swarmathon announced an AprilTags Library object detector would be incorporated in the 2017 base code. The team set aside this project as something to be pursued later for other applications.

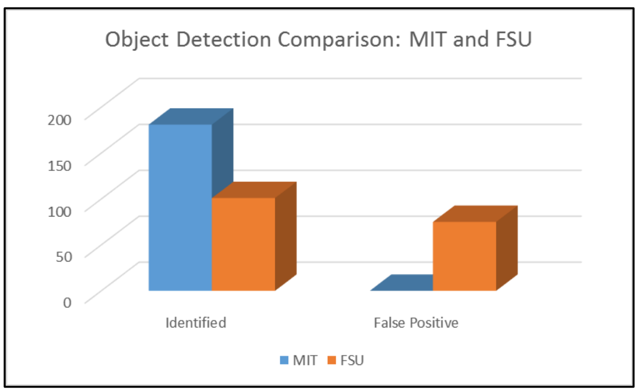

Figure 1. Early testing of Object detector

c) Search area coverage and path planning.

In the early days of our research we wanted to determine if we could improve not just *how* we searched, but *where* we were searching. We hoped to avoid having multiple rovers searching the same area while other areas were unexploited. Informing other robots of a target rich environment was also desirable.

The dissertation by Hoff [5] seemed promising as it discussed foraging, returning targets to a home, and communicating with other searchers. Unfortunately, his method requires some robots that can be dedicated to being beacons that can communicate localization information with the remaining walkers. The number of robots we will be able to deploy is too small to allow us to replicate his method. In addition, given the necessary downward camera focus for cube identification, beacons would not be visualized by walkers but instead avoided as obstacles when detected by sonar.

The strategy described by Mirzaei [6] for including separate search and coverage agents was something we thought we could incorporate, with modifications. As with the beacon issue previously mentioned, we have a limited number of robots to work with and assigning dedicated roles would be imprudent. However, changing their focus at different times during the competition is something we decided we would pursue. This is discussed more under Experiments within *Explore and Exploit.*

In addition, we reviewed the findings of Kolushev and Bogdanov [7] which explored methods that would improve collision avoidance by extending it to include other robots as well. This led us to the concept of rivaling force [8] as a potential means of limiting overlapping search by rovers in proximity.

The concept of rivaling force allows path selection based on a cost function as seen in Fig 2. In our case, we wanted to make an overlap less desirable. At time *k*, rivaling force $F_{ij}(k)$ would be non-zero if two conditions were met. Condition 1 requires that RoverA_2 be within a maximum distance μ and a maximum angle $\pm\Theta$ from Rover A_1. Condition 2 stipulates the difference in their headings is within

$$(180° - \bar{\chi}, 180° + \bar{\chi})$$

with $\bar{\chi}$ representing the maximum allowed difference in heading angle.

Thus, the rivaling force formula (1) will return 0 if Conditions 1 and 2 are not met. K_1 and α are positive constants to be tuned by the investigator, ρ_{ij} is the shortest distance between RoverA_1 and the path of RoverA_2, and $\boldsymbol{\rho}_{ij}$ is a unit vector of the associated normalized partial derivative.

$$F_{ij}(k) = \begin{cases} k_1 e^{-\alpha \rho_{ij}} \vec{\rho}_{ij} \\ 0 \end{cases} \tag{1}$$

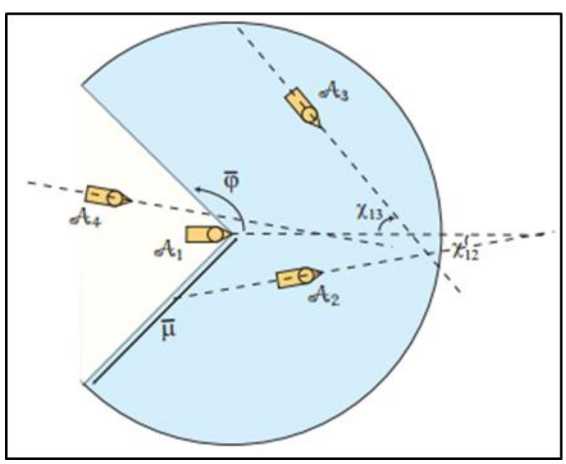

Figure 2. Rivaling forces between agents [8]

III) Process

a) Localization

We had been warned the GPS would not be precise enough to be useful for pinpointing the location of a rover or a target. Testing confirmed this, with significant drift present and results often several meters off target.

Likewise, the wheel encoders had limited usefulness for tracking rover location. Over time, even minute differences in wheel motion, terrain type, and traction not only caused the rover to be inaccurate in predicting its location on the x-y plane, but in reaching a goal location. On a long drive, it would drift from its course, no doubt thinking it was moving in a straight line, but it would eventually produce a curving path. Periodic returns to home to reset *x* and *y* were vital.

The inertial measurement unit (IMU) was reinitialized for each rover both before testing and again closer to the deadline when localization issues during physical testing became increasingly apparent.

b) Rivaling Force

Our simulation results testing rivaling force were promising. When rovers were within the set minimum range of 3 meters and had headings that would result in search area overlap, the rover subjected to the rivaling force would change to a new heading consistently. Physical testing was not implemented for reasons addressed in our Conclusion.

c) Explore and Exploit

i) *Explore:* Phase one of our search and collect algorithm takes place in the first 60% of the time available for the given round. The duration will be determined from the number of rovers deployed, which will also reveal the size of the arena. This information is determined during the start of the competition when the rovers communicate with the host device. From the starting position, rovers will pick a random heading, move 50cm away, and then begin a spiraling search for targets.

ii) *Exploit:* Phase two is scheduled for the final 40% of time remaining of the given round. If a cluster has been identified and published in a global array, and the number of cubes remaining is at least three, any rover still searching at this elapsed time marker will move toward the cluster location. Given localization issues, it may arrive at the coordinates and not see the cluster, in which case it will begin a spiral search in the new area. Ideally, the cluster will be located and the remaining cubes returned. The intent is to leave fruitless searches as the deadline draws near and instead assist in returning targets that have already been identified but not yet collected.

d) Final Search Algorithm

After examining various other search methods, we determined that a spiral-shaped search would be best for the purposes of the competition. It seemed ideal because it would cover a large amount of ground while allowing each rover to easily cover its own separate region. We implemented this by telling the rovers to generally set their goal location to a constant distance to the side, while then adding a variable offset to the distance they travel before that turn. This offset increases over time and resets when it reaches a certain point. Attempting to determine the most effective offset was a major point of focus and made up a large fraction of our testing efforts.

We combined our spiral search with a site fidelity system that allows the rover to remember cube-rich areas and return to them. This was implemented by giving each rover a queue, which was initially empty. The queue would be populated with a location when the rover saw more than three tags, indicating multiple cubes near the rover. The rover will then return to areas stored in the queue, popping them off the queue as it begins its search.

Adding time-based triggers and sharing a global queue were not pursued as originally intended.

e) Gripper Testing

One of our team members with an Engineering focus took responsibility for attaching and wiring the grippers once they were delivered. On one rover, the gripper did not deploy properly until pressure was applied at the pin insertion point. All pins were checked for breakage and appeared intact. The problem remains intermittent.

f) Simulation

We decided to test our code in simulation before running physical tests to identify some of the more obvious problems we might encounter. We quickly determined simulation was of limited use for fine-tuning our code and was best employed to test changes in coarse maneuvering.

(1) *Gripper holds cube in low position:* We noted this and considered it a graphical issue, given that the simulations are GPU intensive. It may be worth addressing at a later date for Virtual teams because we did notice in simulation a cube-gripping rover continuing to engage in search behavior.

(2) *Repeated failure to collect cube:* Multiple rovers experienced problematic cube collection. From observations and testing it was suspected to be due to a communcation delay, wherein the rover decided the attempt was a failure before the blockBlock was registered.

g) Physical Testing

Initial tests were conducted indoors to determine if code changes were being implemented as expected in a general manner. We noticed some of the same behaviors previously noted in our simulations. One rover would select a heading away from home, then begin a search, find a cube, collect it, and turn to face the general direction of Center, and return the cube as expected. A fellow rover would move away from the start point and begin a search, but after collecting a cube an inappropriate heading home is selected. Indeed, in several instances the rover would select a heading directly away from Center.

Another undesired behavior occurred numerous times wherein a rover would return a cube to Center, then cross into Center again after reversing out. It would then find itself trapped and simply spin in place until physically relocated.

A frequent problem involved the rover running over cubes while attempting to collect a cube or while returning one. This was addressed by Swarmathon officials in March by the addition of 'cow-catchers' to the front and rear panels of the rovers.

h) Outreach

In February 2017, our Team Mentor liaised with instructors at two local high schools, Jack Britt and Terry Sanford. Originally 12 students from these schools became involved with the Swarmathon High School Outreach Program: by April 7, ten students remained.

The students were tasked with downloading NetLogo 5.2.0 and importing five modules from a Swarmathon HS Outreach GitHub. The modules guided students, via example, through the construction of small programs that allowed them to build and customize a NetLogo interface as well as write code to direct robots to search for 'rocks' in a given area.

Each module introduced more difficult concepts, introducing students to random walk in module 1, use of pheromone trails in module 2, and Depth-First Search in the third module. By the time they completed module 4 students had also learned about procedures and sub-procedures, site fidelity, and spiral search.

Module 5 provided a template of basic environment and target creation. It was into this template the students would write their own code, using what they had learned in previous modules, to enable six robots to begin at a central location, effectively search an area resembling a planetary hemisphere, collect rocks from single nodes or clusters, and return them to home.

The Outreach team met with students on a series of 2-hour Thursday afternoon sessions starting March 7 and ending April 6, 2007. Most of the students already had some programming experience so they quickly progressed through the modules. At the workshops, members of the internship served as mentors, facilitating critical thinking skills rather than providing answers to the students. Quite often a few leading questions would quickly allow a high school student to answer their own original inquiry, aided by the very clear information and examples provided in the modules.

At the final meeting on April 6, the two teams had two sets of code to evaluate. The Terry Sanford code used a sweeping method like that discussed by Hoff [5], while the Jack Britt team had a separate type of search algorithm assigned to each of six robots. During the workshop, the students added monitors to their interface to allow tracking of the performance of each robot. Multiple simulations were then ran using a variety of rock distributions – some using more clusters, others with mostly single rocks, and of course a few sims with a variable but close mixing of each. The algorithms of the two robots found to be least efficient would be switched with a modified version of the Terry Sanford sweep.

On the evening of April 7, 2017, the final code from the High School Team was submitted to Swarmathon via upload to the Swarmathon-FSU GitHub.

IV) **Results and Discussion**

Our code was effective in simulated testing with both clustered cubes and power law distribution cubes. In the clustered setting, we found that it often took a long time for our robots to discover the large clusters, but once found, they consistently returned to those clusters repeatedly, resulting in efficient cube retrieval. Power law cube distributions resulted in an even more effective search, with our robots both finding smaller clusters of cubes quickly and remembering the locations of the larger clusters. Our results were least effective when there were no clusters of cubes. Without multiple cubes in a tight cluster, our site fidelity was useless and our spiral search had difficulty locating some of the cubes.

While final physical testing closely mirrored simulation results, at the competition we again met with unwanted behavior. As had been noted in earlier testing, we had a rover enter center and become trapped there, while another rover became locked into a spiral. Unfortunately, another rover also pushed cubes off home base by pushing the plastic placard the home AprilTags were printed on from beneath the cubes – this was not deemed to be an environment issue although team members present felt it should have been since the heat had caused the placard to warp upward.

The outcome made it clear we needed to engage in more outdoor testing and aggressively address the behaviors we perhaps carelessly attributed to communication delays.

Our High School team performed well, coming very close to matching the performance of the third place team, but fell behind in efficiency when dealing with clusters.

V) **Future Work**

Future teams can build upon the work and knowledge gathered over the past two years of Swarmathon research. With more physical testing we think the localization problems can be overcome. We have been exploring means by which we can take the rovers away from the interference generated by the buildings and test them in an open area similar to that of the testing arena. Data is needed, however, to determine if the frozen spiral and the unexpected entry into Center can be attributed to camera and communication delays or not.

The Explore and Exploit and Rivaling Force algorithms may prove useful to the incoming team should localization factors be satisfactorily addressed. Much depends on any future changes Swarmathon officials introduce to the 2018 competition.

For our Outreach project, we plan to extend it to include additional high schools. We examined the diversity of numerous schools in the area and decided we needed to select from schools that reflect a greater diversity in their population.

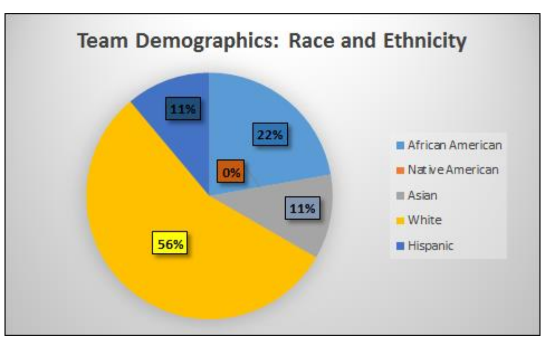

Figure 3. Student race and ethnicity, Workshop Two

We also want to encourage the instructors at the school to extend the offer not only to students they already know are interested in STEM, but to those they feel might become interested if provided an interesting opportunity to learn more. Robotics could be the bridge.

We will also recommend more communication between student team-leaders and the research team members, and will require specific check-ins to confirm acceptable progress is being made. On April 7, we had to do a little "chasing down" to secure the final code for submission: future teams need to avoid that.

Most of the current team will be graduating in the next two semesters, so the future team members facing competition will be able to approach the programming challenges with fresh perspectives while learning from our experiences.

Mr. Lopera has been accepted into the Master of Science in Electrical Engineering program at North Carolina State University.

Ms. Spooner has accepted a summer internship at Lawrence Berkeley National Laboratory working with Protein Modeling.

Ms. Jennette and Ms. Faircloth will work during the summer in a joint program offered by the Center for Defense and Homeland Security and the Office of Naval Research, mentoring high school students in programming in Python and Microsoft Small Basic and the applications of 3D modeling with Unity.

Ms. Gaskins will spend her summer working with the Air Force Reserve Officer's Training Corps in pursuit of her goal of becoming an Air Force Cyberspace Operations Officer.

VI) **Web Links**

Swarm Robotics Internship homepage with links to Project blog and individual blogs:
Swarm Robotics
Project Blog: Swarm Robotics Research Blog
Daphne Faircloth: BroncoSwarm - daphne's blog
Catherine Spooner: My AI Life
Brittany Jennette: My STEM side
Donny Lopera: Swarmathon 2017
Ashlee Gaskins: Gaskins CRA-CREU

VII) **Presentations and Publications**

Object Detection for Target Search in Cooperative Robotics. Poster Presentation. State of North Carolina Undergraduate Research & Creativity Symposium, North Carolina Central University, November 5, 2016

VIII) **Bibliography**

[1] Curiosity Rover, NASA
[2] Swarmathon-ROS GitHub
[3] Kaehler, Adrian and Bradski, Gary. "*Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*." O'Reilly Media, Inc., Sebastopol, CA. September 2015.
[4] AprilTags C++ Library
[5] Hoff, Nicholas Roosevelt, III. "*Multi-Robot foraging for swarms of simple robots*." Harvard University, ProQuest Dissertations Publishing, 2011.
[6] Mirzaei, Mostafa. "Cooperative multi agent search and coverage in uncertain environments." Ph.d Thesis. Concordia University, April 2015.
[7] Kolushev F.A., Bogdanov A.A. (2000) Multi-agent optimal path planning for mobile robots in environment with obstacles. In: Bjøner D., Broy M., Zamulin A.V. (eds) Perspectives of System Informatics. PSI 1999. Lecture Notes in Computer Science, vol 1755. Springer, Berlin, Heidelberg.
[8] M.M. Polycarpou, Yanli, Yang, and K.M. Passino. "A cooperative search framework for distributed agents." IEEE Intelligent Control, (ISIC '01). pp. 1-6. Sept 2001.