Optimizing Object Detection Algorithms Using Ultrasonic Sensors for Robotic Applications

Chris Conrey
Jonathan Dolan
Larissa Gartner
Tyler Lee

ECE 2031, Digital Design Lab
Section L07

Georgia Institute of Technology
School of Electrical and Computer Engineering

Submitted
December 6, 2016

Abstract

In this design report, the team's method to solve a task involving the DE2Bot is explored. The DE2Bot is a robot that is equipped with 8 ultrasonic distance sensors spread at various angles around the ovular shaped frame that is controlled with an onboard FPGA. The task was to autonomously navigate the DE2Bot from a home starting location to objects placed randomly throughout a predetermined size field. Upon finding an object, the DE2Bot should navigate to and touch the object then proceed to return back to the home location. The successful completion of this process constitutes an object tag, and the object is subsequently removed from the game board. In order to accomplish these tasks, the team broke the problem down to three fundamental pieces - search, tag, return. This report delves into the process of developing an algorithm based around robot odometry and the distance sensors to independently solve each of the these three tasks, as well as the lessons learned from testing the algorithms.

Optimizing Object Detection Algorithms Using Ultrasonic Sensors for Robotic Applications

Introduction

Robotic automation has seen a relatively recent surge in popularity and as such has become a quickly expanding sector in the global capital marketplace. The increased use of automation to optimize manufacturing and related processes has generated the demand for accurate, repeatable odometry and object detection. In warehouse applications, companies such Amazon utilize a logistical robotic approach to perform tasks in an efficient manner. Upon successfully identifying products on the warehouse floor, robots must maneuver to the object before subsequently delivering the product to a base location which requires precision odometric sensors with the ability to maintain their accuracy. This report explores the process of identifying and tagging a variety of objects scattered throughout a finite area and overcoming periodic odometric error in the process.

Overview of Approach to Accomplish the Goal

The specific scenario researched relies on a DE2bot discovering obstacles in an 8' x 12' arena, coming into contact with each placed obstacle, and returning to the starting position. This task was accomplished via programming the Simple Computer (SCOMP) operating on the FPGA integrated in the DE2Bot. In order to effectively accomplish the task, streamlining the turn and tagging sequence and maximizing the usage of onboard ultrasonic range sensors were development priorities.

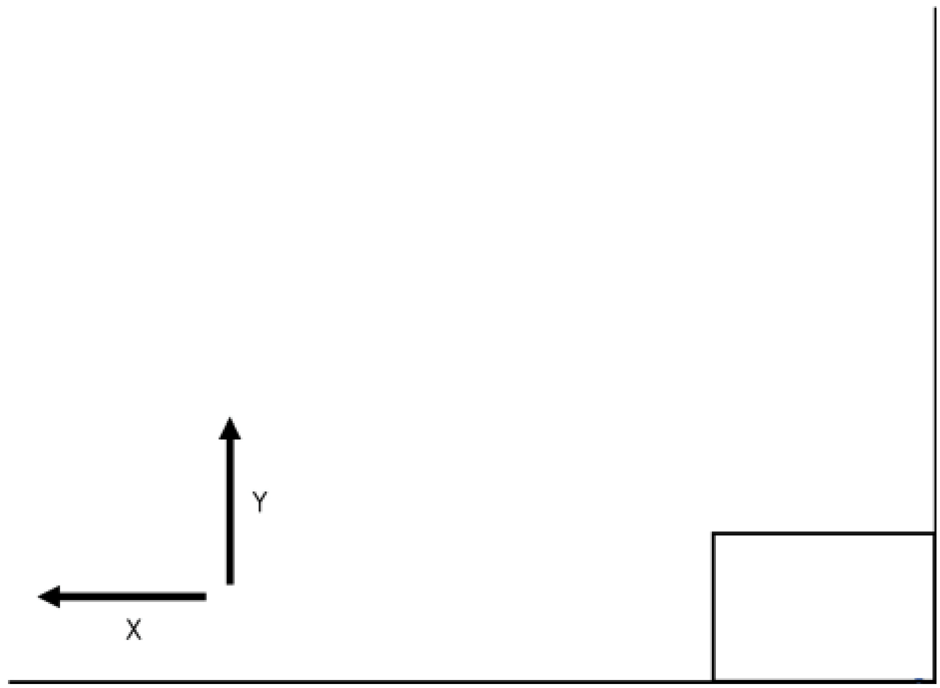

Figure 1. The arena, 12 feet in the X direction and 8 feet in the Y direction, with the home square placed in the bottom left hand corner, measuring 2 feet by 2 feet.

The implementation was broken into 3 main functions: robot seeking, tagging, and returning. For seeking, the team developed an algorithm referred to as the “lawn mower” approach where the robot travels up the game board, moves slightly across, and travels back down the board. For tagging, the robot follows the algorithm referred to as the “right turn tagging method.” This simplistic but effective approach instructs the robot to turn right 90 degrees when an object is detected and approach the object until contact can be confirmed. The final key aspect of the approach is the returning function which brings the robot from the tagged object into the home square, defined as a 2 foot square with the bottom left corner as the lower left extent of the game board, as depicted above in Figure 1.

General Methodology

The approach the team took towards solving the problem remained malleable throughout the development process in order to utilize capabilities as they were successfully implemented, avoiding time consuming approaches. The final methodology was reduced in complexity from many of the algorithms referenced in the design proposal, including sinusoidal or zigzag search paths and dynamically mapping object positions to optimize finding the next object after returning home. The final approach was most similar to the contingency plan, which included a large amount of human interaction to reduce propagation of odometry error, and did not utilize a search method of any kind. This was due to problems encountered during development, which are detailed in the sections below. Each main function required for the complete algorithm to be successful is discussed in detail, as well as the problems encountered and resulting final demonstration.

Development of Search Path

Development of the searching path algorithm underwent numerous iterations that simplified the DE2Bot's movement path and the code that implemented it. In the proposal, the benefits of various searching algorithm approaches were discussed. These patterns would thoroughly scan the entire arena, accounting for objects that are more difficult to detect, such as cylinders or non-reflective cubes. However, these more specialized search paths, such as a sinusoidal or zigzag sweep, see Figure 2, covered the arena more thoroughly than necessary. As the design progressed, it became clear that the complexities of these paths was not worth the efficiency tradeoff, especially given the relatively confined arena dimensions. These realizations lead to the team's final searching algorithm which traversed the DE2Bot along the longest dimension of the arena, traveled half of the shortest dimension, and then made its way back down the longest dimension, referred to as the "lawn mower" approach illustrated by Figure 2.

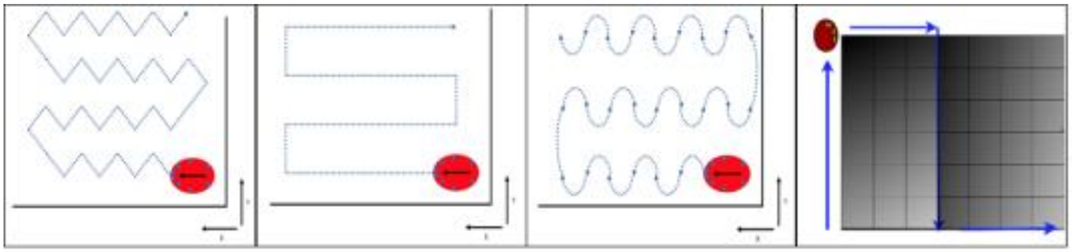

Figure 2. (a) Zigzag Path (b) Rudimentary Lawn Mower Path (c) Sinusoidal Path (d) Finalized Lawn Mower Path

The idea of dividing the arena in two stemmed from tested reliability of the ultrasonic distance sensors. While the specifications of the ultrasonic range sensors detail a range of six meters, beyond the confines of the arena along the shorter dimension, farther objects placed in harder-to-detect orientations could still go undetected. By following a simple linear path down the X axis of the game board, the DE2Bot would be able to detect any favorably oriented object in the arena without needing to turn and travel back down the middle of the arena. However, there is still a possibility of distant objects with troublesome orientations or shaped going undetected. Therefore, returning down the middle of the arena with active left and right ultrasonic range sensors eliminates the potential to miss objects without significant additions to tagging time, shown in Figure 2.d.

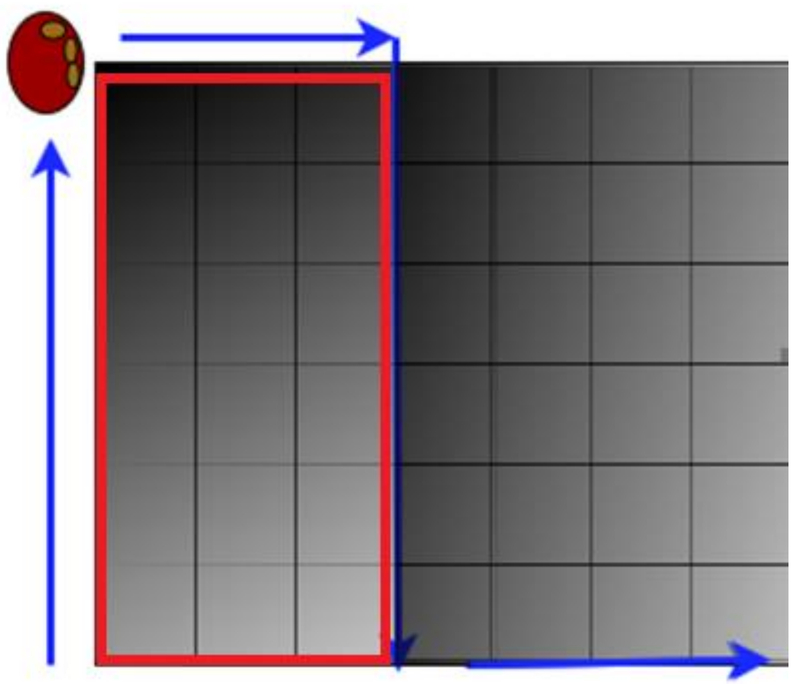

Figure 3. The lawn mower approach in more detail, with the red box indicating the area guaranteed clear by the first pass down the game board.

Deviations from the lawn mower approach in the final demonstration

The final code produced for the DE2Bot demonstration employed a single linear sweep search path. In this approach, the DE2Bot only traveled the longer side of the arena while detecting objects and did not turn right at the end of the board to make a sweep back towards home. This approach was adapted for the demo due to time constraints, but would not be the strategy the team would use going forward. The technical difficulty of adding the return sweep is not beyond any of the capabilities the team successfully implemented, making it an easy addition to the code.

Right Turn Tagging Method

The implementation of the seeking algorithm that the team decided on was a compromise between effectiveness and ease of implementation. The technique, decidedly called the "Right Turn Tagging Method", utilized the right facing sonar sensor of the DE2Bot as it travels forward from the home area along the long wall of the arena. When the sensor detected an object, it was presumed to be 90 degrees to the right from the front of the robot, so the DE2Bot then turns 90 degrees to face the detected object.

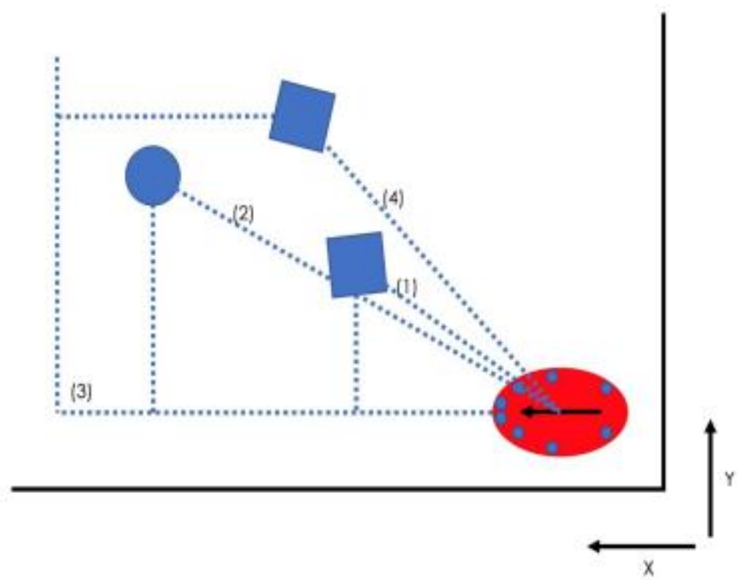

Figure 4. Path that the DE2Bot follows in the Right Turn Tagging Method. The robot travels down line 3, and when an object is detected it executes a 90 degree right hand turn.

However, The DE2Bot ultrasonic sensors output signals that propagate in a cone shape with an approximately 20 degrees of separation from the left and right edges. This means that the right facing sensor can detect an object approximately 10 degrees before it is actually directly in front of the sensor.

To compensate for this premature detection, the front ultrasonic sensor angled 12 degrees to the left is utilized for object honing after the robot executes its turn. This allows for the error introduced by turning before the object is exactly 90 degrees to the right to be minimized, shown in Figure 5.

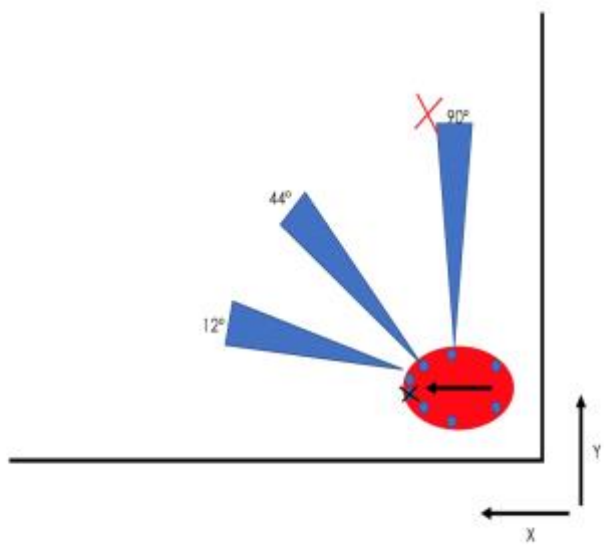

Figure 5. Depiction of the DE2Bot detecting an object at the location indicated by the red X, which is noticeably before the object is exactly 90 degrees to the right of the robot or directly in front of the right facing distance sensor. Utilizing the left front facing sensor on the robot, marked with the black X, allows for this effect to be compensated for then the robot executes its 90 degree right turn.

The DE2Bot then traveled forward until the measured distance was just above the measurable threshold of 15 cm. At that point, the DE2Bot would travel 17 cm forward and presume it had successfully touched the object. Finally, the DE2Bot would transition into the return home method.

Problems in the implementation used in the final demonstration

The Right Turn Tagging Method was implemented in the code used for the demonstration, but it did not function correctly. During testing, the DE2Bot consistently turned to face objects detected on its initial outward sweep. As the robot moved forward, relying on the front facing sensors, it wasn't able to correctly identify the presence of an object due to the lack of a directly forward facing sensor. This led the DE2Bot to continue driving forward, even after making contact with the object. The team believes utilizing the right front facing sensor in addition to the left sensor would be beneficial to solving this problem. By combining these two distance measurements, the DE2Bot would be able to see a much wider arc in front of itself, allowing for inaccuracies in the turning to be further minimized. In this implementation, however, the risk for objects that aren't being targeted to become an error inducing factor grows. Many edge cases would need to be tested after implementing this approach, but it could provide a promising improvement to the algorithm.

Return Home Method

The return home method describes the process the DE2Bot would take to travel to the home area after touching an object. The DE2Bot would start by reorienting itself towards home and proceed to move in a straight line until it reached the origin. To implement this, the program would utilize the provided ATAN2 method to calculate the angle required to face home based on the DE2Bot's current Cartesian coordinates provided by odometric sensors. The DE2Bot would turn in place to this angle, and begin to move in that direction. During movement, the DE2Bot would constantly reevaluate the theta required to return to the origin, again utilizing the ATAN2 function. This functionality was included to ensure the DE2Bot moves towards the origin even as error begins to accrue. The updates to the heading after the initial turn are executed using the included movement API, which means the DE2Bot turns as it moves as opposed to turning in place. The movement loop would be terminated when the DE2Bot's odometry reports it is on the borders of or in Cartesian quadrant three. Upon reaching this terminating condition, the DE2Bot turns itself in place to a heading of 0 degrees and repeats the entire control loop.

Deviations from the return home method in demonstration implementation

The implementation demonstrated involved a few simplifications from the aforementioned approach. Instead of rotating to an angle calculated by ATAN2, the DE2Bot instead executed a left 180 degree turn to position itself facing the long side of the arena along which it started moving, as shown in Figure 6.

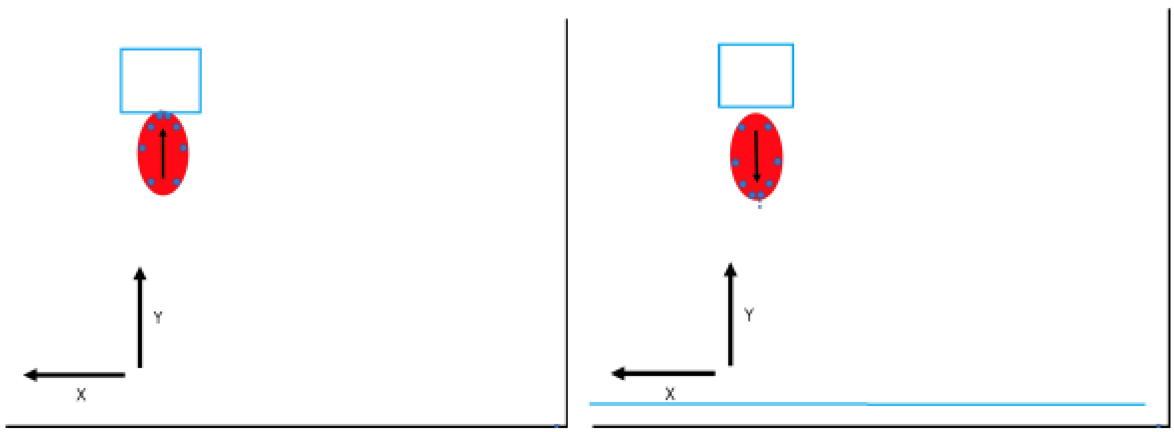

Figure 6. The left image shows the DE2Bot as it makes contact with a detected object, and the right image shows the DE2Bot after it has executed its left 180 degree turn to face the line on which it traveled out on, shown in blue.

At this point, the DE2Bot moves in a straight until its Y translation odometry value is 5 centimeters. The DE2Bot then executed another left turn, this time 90 degrees, to orient itself towards the home area. The DE2Bot then moved forward until its X odometry value read 5 centimeters, at which point it assumed it was in the home area.

Deviation from the planned implementation was not intentional, but was instead dictated by the necessity to code a more suitable approach for the demonstration, given some missing functionality. This approach proved to be significantly less complicated, utilizing capabilities the team was able to previously demonstrate. There were significant downsides to this implementation, most importantly that the path was no longer the shortest possible, thus becoming less time-efficient . In addition, this approach left the DE2Bot at a coordinate (5, 5) centimeters, which did not guarantee that the DE2Bot would be entirely within the home area. The (5, 5) centimeter ending point was included to prevent wall collisions, which would cause worse penalty than a retrieval. However, this approach did prove to be beneficial as well. The path the DE2Bot would take to return home is equivalent to the path the DE2Bot took to get to the object.

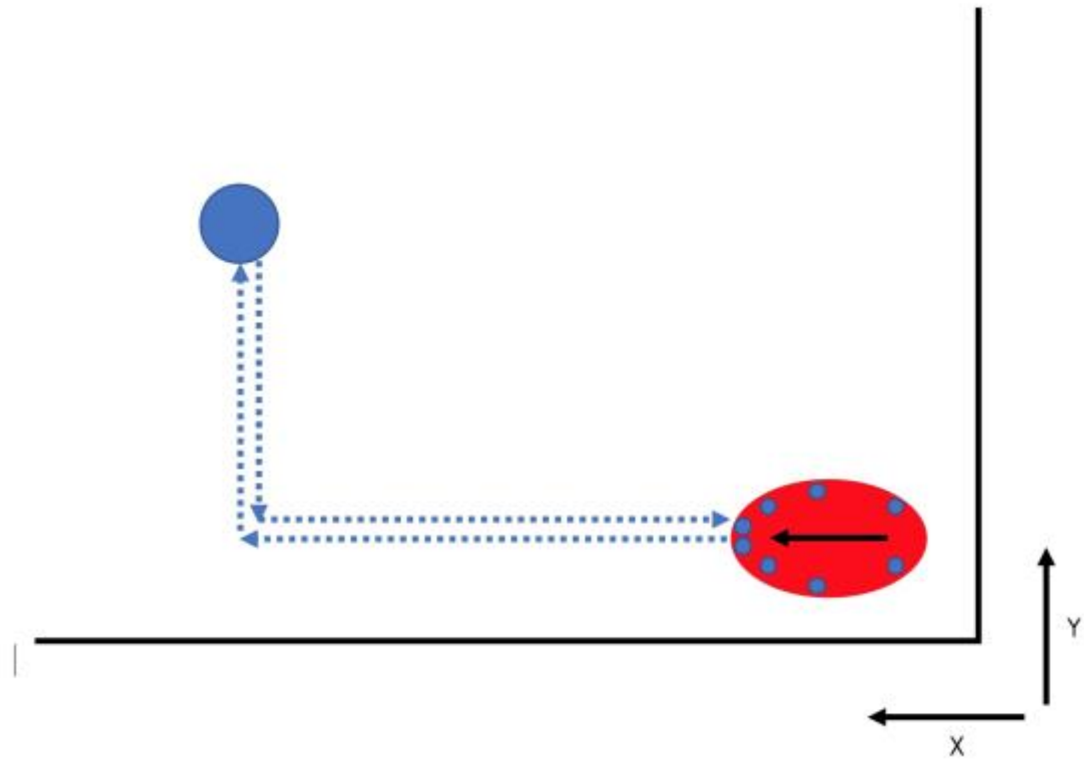

Figure 7. Return home path modification which inverts tagging sequence and requires user realignment in the home region

This ensured that the DE2Bot's path home would be unobstructed, therefore minimizing the risk of a collision while traveling home. Also, since turns in multiples of 90 degrees was a proven to be accurate, using only 90 degree turns reduced the complexity of the entire approach and matched the implementation of the searching algorithm.

Project Management

The project was divided into two major components, the coordinate navigation of the DE2Bot and the ultrasonic distance detection of obstacles with each portion containing modular subsections to simplify debugging. The majority of the time was exercised determining how to control the DE2Bot movement and maintaining control of positional variables for future use in the return home routine. The ultrasonic detection sequence was developed after fundamental movements had been finalized as a method to switch between the movement subroutines. A revised project timeline of the actual realization is depicted in the Gantt Chart in Appendix A.

The work distribution was fairly homogenous with the goal being for each member to understand all elements of the project and thus be able to contribute in all aspects. Each of the members wrote code, debugged elements, and worked on the documentation with a slight polarization between developers and writers. The main working periods were predominantly collaborative coding with most of the documentation written in external meetings.

Technical Results

The final results varied significantly from the aim of this design. In two runs, the DE2Bot was unable to successfully complete a full cycle of seeking, tagging, and returning any objects home. The robot sought out and tagged 5/10 of the objects placed during the two runs. The DE2Bot did not successfully return home on its own accord, which led to a total of 13 retrievals. There was an intermittent issue of the bot making a 90° right hand turn every time the program loop restarted, which was caused by the ultrasonic sensor immediately discovering a non-existent object and entering the right turn tagging method initially. Although the final score was 590, the DE2Bot code did not perform up to the team's standards due to the inconsistent results and lack of autonomy.

Table 1. Demonstration Results

Run	Tags	Collisions	Retrievals
1	2	0	5
2	5	0	8

Conclusions

The intended goal was never successfully accomplished as a result of the numerous barriers encountered in the design process. Despite functioning turning techniques, coordinate based control flow, and ultrasonic detection project elements, the final demonstration with the conjoined code illustrated only an error-correcting right hand turn and straight movements, as opposed to the outlined plan of detection, tagging, and returning home.

Positive Takeaways

Ideally, the devised code would have functioned modularly for quick adaption to a wide variety of scenarios. By taking advantage of three ultrasonic sensors mounted at various angles and developing corresponding subroutines to turn at those particular angles, object detection was simplified regardless of orientation. Simple modification of the designed code also permitted the usage of the three front facing rangefinders not utilized, could have allowed easier detection.

Shortcomings

The robots homing techniques are a definite area for improvement as the return home currently solely relies on the coordinate input with the rangefinders providing no additional support for precision despite the homing region being surrounded by wall. Working with CALL and RETURN functions proved to be challenging, leading to the code tested in the demo utilized only JUMP functions. Although this code functioned, working with CALL and RETURN should provide a more efficient technique within the VHDL code.

Future Upgrades

One of the major proposed optimizations is the chaining of objects; if another object is detected during the tagging stage, the secondary object's location could be recorded or the last known DE2Bot location where a subsequent object had been found. This would provide considerable time savings as the DE2Bot would no longer need to restart the programed path upon returning home, rather opting to go straight to retrieving the next object. Another future modification is the expansion of the code to cover a rectangular region of any size, especially since the ultrasonic range finders specifications were only adequate enough to cover the objective region on a single pass. An expansion of the lawn mower approach would function perfectly in any sized rectangular region as nearby object are cleared before further objects would be collected, avoiding any potential collisions.

A final recommendation to future students would be to spend less time devising the simplest approach, allocating more time to adjust tracking techniques. Proper time management along with following preset team deadlines is vital.