AUTONOMOUS SEARCH AND RETRIEVAL

Sage Kleczynski, Matthew Rayner and Suranga Hettiarachchi
Computer Science Department
Indiana University Southeast
4201 Grant Line Road,
New Albany, IN 47150
Email: {sagemk@gmail.com, raynerm@ius.edu, suhettia@ius.edu}

ABSTRACT

The purpose of our research is to develop an autonomous robot algorithm that utilizes sonar sensors. The algorithm allows the robot to navigate through an obstacle laden static environment to a predefined location where robot retrieves a desired object and then navigate back to the starting location. The robot utilizes data from the sensors to update a map of the environment and chooses the best path to the goal point based on multiple path determination algorithms. Success of the algorithm is based on successful navigation to the goal location, and navigating back to the starting point, hence minimum error.

Keywords **–** autonomous navigation, search and retrieval, obstacle avoidance

1. INTRODUCTION

Our research objective is to develop an algorithm for a mobile robot to autonomously navigate a static environment with the intention of reaching a specified goal location on a map while avoiding obstacles. Upon reaching that location, have the robot retrieve a desired object, such as picking up a physical object or collecting data of the environment. Finally, the robot returns to the starting location to deliver the object or data.

There are various real world scenarios that require humans to conduct repetitive and sometimes dangerous tasks for which this could be applied. For example, first responders may have to transport supplies through unknown and potentially hostile environments on a daily basis. Development of a method for a robot to retrieve and transport these supplies in the place of humans could save countless lives and free human resources to be used elsewhere. Currently the US military is in fact working on a project, *TerraMax*. The TerraMax is a self-driving truck that is capable of safely transporting supplies, and possibly even soldiers in high-risk areas. It uses radar and light detection and ranging (LIDAR) to generate a map of its surroundings and navigate through unknown environments [2].

Hazardous environments are another situation in which autonomous search and retrieval could be applied. Radioactive or chemically contaminated environments pose a great threat to humans in that they are not visibly noticeable dangers and one can be exposed to them without knowing. Instead of humans, robots could be sent to investigate and eliminate these harmful contaminations. One such example is the use of robots to monitor radioactivity at Fukushima Daiichi Nuclear Power Plant in Japan after the March 2011 earthquake [5].

2. BACKGROUND

There has been a significant amount of study done in regards to autonomous navigation and planning algorithms in the field of robotics. Some of these studies offered a lot of useful knowledge and ideas in the design process of our algorithm. Two particular methods we looked at were Autonomous Navigation with Hazard Avoidance (AutoNav) and the use of a vector field histogram (VFH).

2.1 AutoNav

AutoNav is a navigation algorithm that was implemented on the Mars rover and is based on the GESTALT (Grid-based Estimation of Surface Traversability Applied to Local Terrain) algorithm [3][6]. AutoNav uses cameras to evaluate its surrounding environment and then uses that information to generate a model of the terrain. Part of this model is a grid-based representation of the environment called a goodness map. Each cell in this grid contains a goodness value that indicates the difficulty of traversing that cell. A high goodness value indicates that a cell is easily traversable and a low goodness value indicates that a cell is hazardous. If a cell in the goodness map is considered to be hazardous then it is expanded in all directions by the rover radius, which creates a buffer zone to account for the robots size and allows the robot to be treated as a point on the map. Once the terrain has been evaluated the robot then considers a set of possible arc shaped moves from its current position. Each possible move is evaluated based on the following three criteria: hazard avoidance, steering time, and destination location relative to the goal. A vote is then generated based on these criteria and the rover chooses the best move. It moves a predetermined distance and repeats the process. AutoNav works well for keeping the robot away from hazards but algorithm fails when the hazard density is high causing the robot to get stuck in cul-de-sacs.

2.2 VFH

One key component of an autonomous navigation algorithm is the ability for the robot to avoid obstacles. When dealing with ultrasonic sensors, it is important to have some redundancies in the algorithm to account for noise in the environment. In our research we found this can be accomplished by using a vector field histogram grid as a model for the environment [1].

A histogram grid is generated by rapidly firing multiple sensors around the robot during motion. For each distance reading returned, the cell in the histogram grid that lays in the direction that particular sensor is scanning and is equal to the measured distance is incremented. As this process continues, some cells in the grid will continue be assigned a larger and larger value indicating the probability of that cell containing an obstacle. Other cells that the sensors may have acquired an inaccurate reading on will have much lower values. This creates a "certainty" map. A threshold value is then compared to the cells and anything above that threshold is considered to contain an obstacle and anything below that threshold is considered to be open. The result is a much more accurate depiction of the environment.

3. PROJECT DESCRIPTION

Our hardware platform is an X80Pro robot developed by DrRobot [4] (see Figure 1). The general steps of our algorithm are to scan the area, map the data, and navigate through the obstacles mapped. While navigating, the robot continuously checks the cell ahead for any obstacles that were not previously detected. If a new obstacle is detected, the agent repeats the scanning and mapping before navigating.

3.1 Scanning

During the scanning phase, the robot resets to a Northward bearing. It then rotates 360 degrees taking readings from 6 sonar range sensors (3 in the front and 3 in the back) every 5 degrees. (Figure 1) This equates to a total of 432 readings with 6 readings in 72 directions.

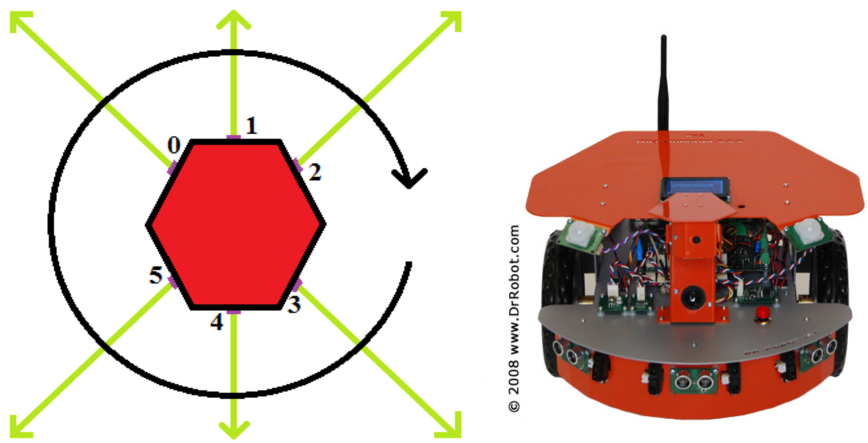

Fig 1: showing sensor positions of X80Pro robot.

These readings are stored in a two dimensional array. This array can be represented as a histogram (Figure 2) by removing any values outside the accepted range (0.08 – 2.55 m). This array is used during mapping to add obstacles to the current map. When the scanning is repeated the array is cleared and a new array of readings is taken.

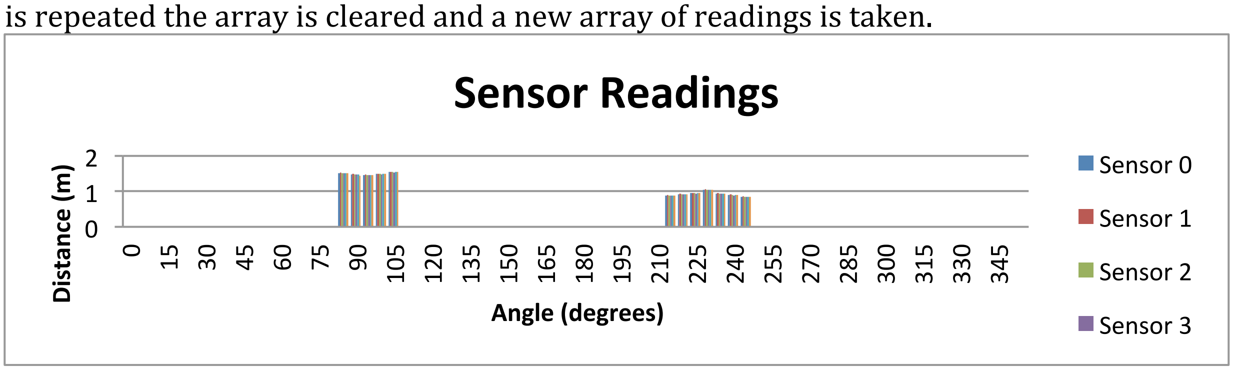

Fig 2: Histogram of sensor readings of the accepted range of values.

3.2 Mapping

Once all of the data is gathered from the sensors, the robot switches to mapping mode. Once in the mapping mode, the robot uses the histogram to update obstacle locations on the map. The grid-based map is represented as a two dimensional array where each cell represents a $10cm^2$ area of space. The instance of the robot itself in the system stores its current location within the grid. Given the relative x and y coordinates of the robot and the sonar range sensors at each of the 72 bearings, we can calculate the relative obstacle locations using, **$x2 = x + (range * \cos(rad))$** and **$y2 = y + (range * \sin(rad))$.** Where x and y are the current robot location, range is the distance to the obstacle in cm, and rad is the bearing in radians. Then x2 and y2 is the position of the obstacle.

Once the location of the obstacle is determined, the area covered by an obstacle is updated on the map. We use a "certainty map" where an obstacle exists in any given $10cm^2$ cell. For each location determined by (x2, y2), we move out two cells in every direction. This 20 cm increase in the size of the obstacle helps account for the 40cm diameter of the robot itself. For all 25 cells, we increment the certainty value in that cell by 3. The certainty map is persistent so the values do not reset on subsequent rescans. The cells have a maximum certainty of 150. If any cell value goes above this threshold, value is simply reset to 150.

The left picture of Figure 3 shows the certainty map where the yellow square represents the current location of the robot. The blue cells are empty, and the red represents higher certainty values.

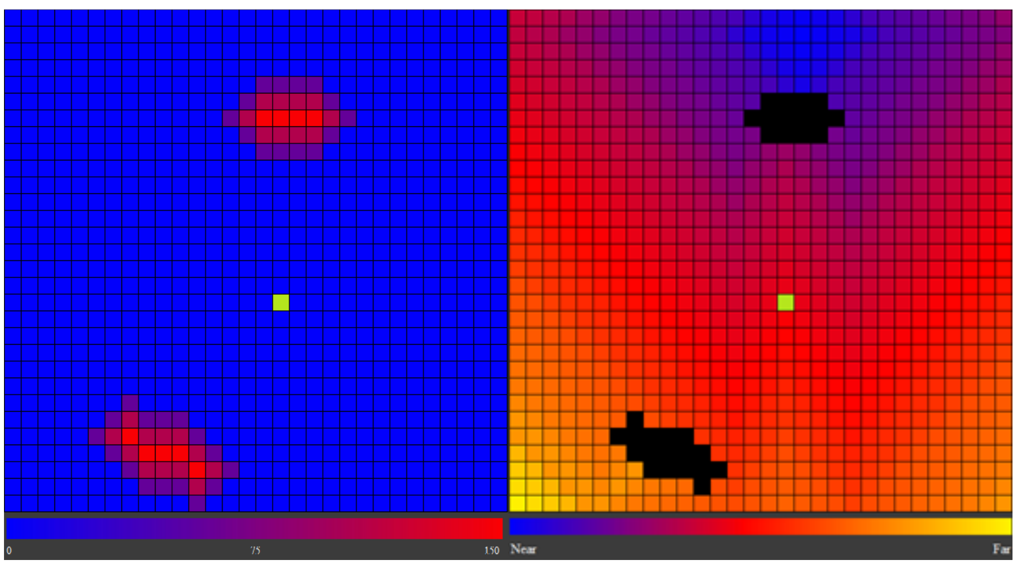

Figure 3: Certainty Map (left) and Goodness Map (right)

3.3 Navigation

Using the certainty map as a reference, we generate a goodness map for navigation (right picture of Figure 3). The goodness map is a two dimensional array with the same dimensions as the certainty map. Through observations we settled for goodness value of 70. Therefore we consider any cell with a value above 70 to be non-navigable.
Once the goodness map is generated, we use a propagating wave front algorithm to find the shortest path. This algorithm starts at the designated goal location on the goodness map and sets its value to 0. This represents the distance to the goal. Then the algorithm increases the local distance variable and move in each of the four directions that are both not occupied by an obstacle, and either have a current distance value greater than the local

distance variable value, or is null. This generates a suitable navigation path to start moving towards the goal but it generates a "*zigzag*" motion. To avoid this motion, we introduced weights to the cells depending on the current bearing of the robot. The cell directly in front of the robot gets 0 weight, and the three adjacent cells get a weight of 1 to their distance values. This allows the robot to travel straight making the robot takes unnecessary turns.

The goodness map, unlike the certainty map, is not persistent. It is completely rebuilt every time the certainty map is updated. This ensures that the robot does not run into obstacles that were not present in the previous iteration of the map.

4. RESULTS AND ANALYSIS

To test our algorithms, we used three different test scenarios. All data is averaged over 10 runs. The three test scenarios are shown in Figure 4, the green lines represent the nominal path taken to the goal, and the red line represents the nominal path back to the home location.

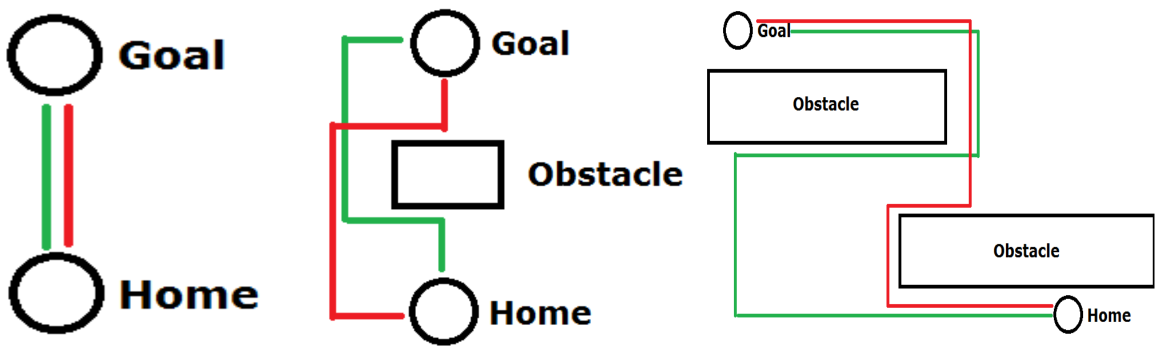

Figure 4: Three test scenarios-no obstacles (left), single obstacle (center), and two obstacles (right)

4.1 No Obstacles

In the first scenario, the goal is placed 1 meter directly in front of the robot's starting location (Figure4 left). There were no obstacles placed in the robot's path. The results show an average horizontal error of 0%, vertical error of 2.4%, and angular error of 0%. The non-existence of any horizontal error is accounted for in the fact that there is no horizontal movement of the robot. The 0% angular error is due to the robot only turning 3 times, the two scans at the beginning and after reaching the goal, and the 180 degrees turn to return home after reaching the goal.

4.2 Single Obstacle

The second scenario includes an obstacle between the robot's start location and the goal (Figure4 center). The results show an average horizontal error of 0%, vertical error of 2.4% and an angular error of 0.09%. In this test, the westward horizontal movement equaling the westward movement, the total horizontal movement becomes 0 with no error. The angular error is due to robots taking turns to avoid the obstacle.

4.3 Two Obstacles

In the third scenario, we placed two long boxes roughly a meter apart, but at opposite sides of the map to only overlap by a small amount. This placement caused the robot to travel the greatest distance, with a large number of turns, and rescans for unknown obstacles. The results show a larger average error in all measurements with an average vertical and horizontal error of 1.8% and an angular error of over 0.2%. We can observe a larger variance from the previous two scenarios where the goal has no horizontal shift from the home location.

5. CONCLUSIONS AND FUTURE WORK

We were able to successfully develop and implement autonomous search and retrieval algorithm on a physical robot. Our algorithm uses six sonar sensors to generate a certainty map and a goodness map of the environment. The robot is able to determine the shortest path to a goal position, safely navigate to goal position while avoiding obstacles, capture a picture at the goal position, and finally return to its starting position.

Overall, We are reasonably confident that our approach worked quite well in small scale. Although more testing would be needed; it is apparent that our algorithm scales up with larger and more complicated static environments.

In our future work, we expect to expand the mapping algorithms to not only increase the certainty values in cells where obstacles are located, but to also decrease the values in all cells between the robots current location, and the scanned obstacle. This would lead to a better knowledge of the environment allowing better path planning. We would also enhance the calibration of the robot itself, to more accurately turn the robot in more precise movement. With this we would also be able to explore moving in more fluid dimensions rather than the four cardinal directions.

REFERENCES

[1] Borenstein, J., The Virtual Force Field (VFF) and the Vector Field Histogram (VFH), IEEE Transactions on Robotics and Automation, 7(3), 278-288, 1991.

[2] Braid, D., Broggi, A. and Schmiedel, G., The TerraMax Autonomous Vehicle, Journal of Field Robotics, 23(9), 693–708, 2006.

[3] Cartsen, J., Rankin A., Ferguson, D., and Stentz. A., Global Path Planning on Board the Mars Exploration Rovers. Proc. of the IEEE Int. Conf. on Aerospace, 1-11, 2007.

[4] Dr. Robot., Inc. X80 WiFi Mobile Robot Development Platform PRO Version. http://www.drrobot.com/products item.asp?itemNumber = X80pro, 2009.

[5] Kawatsuma, S., Fukushima, M., and Okada, T., Emergency Response by Robots to Fukushima-Daiichi Accident: Summary and Lessons Learned, Industrial Robot: An International Journal, 39(5), 428 – 435, 2012.

[6] Maimone, M., Johnson, A., Autonomous Navigation Results from the Mars Exploration Rover (MER) Mission, in Experiment Robotics IX, Springer Tracts in Advanced Robotics, 21, 3-13, 2006.