|  |  |
| --- | --- |
|  | Rose-Hulman  Institute of Technology |

Memo

To: Dr. Carlotta Berry

From: Ander A Solorzano \_\_\_\_\_\_\_\_\_\_\_\_\_\_ and Ruffin White \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: ECE425 – Mobile Robotics

Date:

Title: Lab06 – Path Planning (Occupancy Grid and Topological)

**PURPOSE**

The purpose of this lab is to use topological and metric navigation to move the mobile robot (CEENBoT) from a start point to a desired goal location. Topological path planning is based upon landmarks in the world where distinctive places such as landmarks or *gateways* help the robot localize itself in the world and trigger actions that might change its direction. Metric path planning involves the use of a *wavefront* algorithm on a priori map to create a path from the robot’s start position to the goal location. The final task of the robot is to demonstrate world navigation from a start point to and end point using *wavefront* expansion (i.e. metric navigation) on a topological map of the world’s salient features (topological navigation). The list of robot commands and/or generated *wavefront* should be shown on the LCD to make it clear about the robot’s current state and next state.

**PROCEDURES AND STRATEGY**

Before we began coding the robot to handle topological and metric navigation, we acquired some IR readings about different landmarks and gateways. With these readings, we where able to set some threshold values that would check each of the gateways and allow the robot to trigger a necessary change in direction.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Landmark** | **Front IR** | **Left IR** | **Right IR** | **Back IR** | | Corner in Front | 15.3 | 23.4 | 22.7 | 50 | | Corner on left | 7.8 | 10.3 | 50 | 50 | | Corner on right | 7.8 | 50 | 10.5 | 50 | | Hallway on both sides | 50 | 10.2 | 9.8 | 50 | | Hallway on left | 50 | 50 | 10.1 | 50 | | Hallway on right | 50 | 10.4 | 50 | 50 | | T-junction | 8.1 | 50 | 50 | 50 | | Dead end | 8.2 | 10.8 | 10.2 | 50 | |
| Table 1: IR sensor readings for topological landmarks present in the robot world. A reading of 50 was given to values that were too large and too noisy to acquire a clear reading. The readings are in centimeters and NOT in inches. |

Now that we have acquired landmark readings of the robot world, we procedeed to create the topological and metric path planning behaviors of the robot. Some values might vary due to systematic errors from the sensors and nonsystematic errors from the environment.

Topological Behavior:

We started by implementing a user interface where the user will be able to input *turn left, go straight,* and/or *turn right.*The reason for doing this is so that we can tell the robot to “turn on the second left”or “turn on the third right” for example. Rather than telling the robot to “turn on the first available right” or “turn on the first available left,” we decided that this method is more user-intuitive sinc it would relate to telling instructions to a person. For example, these instructions would include: turn left at the corner, continue walking down the hallway until the third right, turn right and walk down until you reach your destination. After we coded the user inputs, we incorporated to implement our landmarks into the robot map. This way the robot would always be aware of what obstacles to expect on the current cell and on the next cell. A side feature that we included to our topoligical behavior consisted of a starting orientation setting. We would be able to indicate whether the robot starts by facing North, East, South, or West and the robot would correctly orientate itself to start moving in the map.

Metric Behavior:

We incorporated the wavefront diagram of the world using 4 bits for each of the 16 cells. The MSB correlates the front IR sensor, the right IR sensor correlates to the 2nd MSB, the back IR sensor correlates to the 3rd MSB, and the left IR sensor correlates to the LSB. In other words, a sensor reading of 0b0101, or 5 in decimal, would indicate that the robot is on a hallway with walls on both sides. The main feature about our wavefront, is that we are able to input a initial starting point and the final goal location. The robot would then traverse along the map using the map of the global map of the wolrd to traverse fix distances.

Metric Behavior with a Topoligical Expansion:

Once the two basic behaviors were coded and implemented, we then proceeded to create a more complex path planning behavior that incorporates metric and topological path planning features. Before launching the robot, we have the enter the starting and end positions for the the *wavefront map.* However, we must also put in the desired user move commands such as turn left, turn right, go forward, and the starting orientation of the robot. The most difficult aspect of this behavior is that we must be able to always store and distort the wavefront map when the robot turns. This is because when the robot turns, the wavefront map also changes according to the robot’s from view. This is similar to a GPS since the most popular types of GPS have the car marker always face forward while the world rotates around it. To implement this difficult yet uniqure feature into our code, we used a barrel shifter.

**QUESTIONS**

1. What was the strategy for implementing the wavefront algorithm?
2. Were there any points during the navigation when the robot got stuck? If so, how did you extract the robot from that situation?
3. How long did it take for the robot to move from the start position to the goal?
4. What type fo algortihm did you use to select the most optimal or efficient path?
5. How did you represent the robot’s start and goal position at run time?
6. Do you have any recommendations for improving the robot’s navigation or wavefront algorithm?

**CONCLUSION**