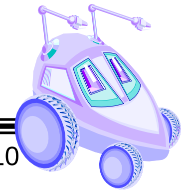


Lab 7 Metric Path Planning (Occupancy Grid)

Reading: *Introduction to AI Robotics (Ch. 10), Lectures 6-2, 7-1*

(Demonstration due in class on *Thursday*)

(Code and Memo due in Angel drop box by midnight on *Thursday*)

Purpose: The purpose of this lab is to implement metric path planning by using a wavefront or grassfire expansion on a metric map to move the mobile robot from a start point to a goal point.

Equipment: Base Robot
IR/Sonar Sensors
Servo Panner

Software: Microsoft Visual Studio.NET 2008 with C#
Serializer.NET library and firmware
Bluetooth transmitter

****

LAB PROCEDURE

****

Metric Map Path Planning and Execution

1. Use a wavefront algorithm on an a priori map to create a path from the robot's start position to goal location. Use the obstacle avoidance and move to goal behaviors to move through the list of goal points until the robot arrives at the final destination. Assume that the algorithm uses an eight-neighborhood so that the robot can move diagonally. The test arena will be 6 ft x 6ft with 9" x 9" obstacles (see Figure 1).

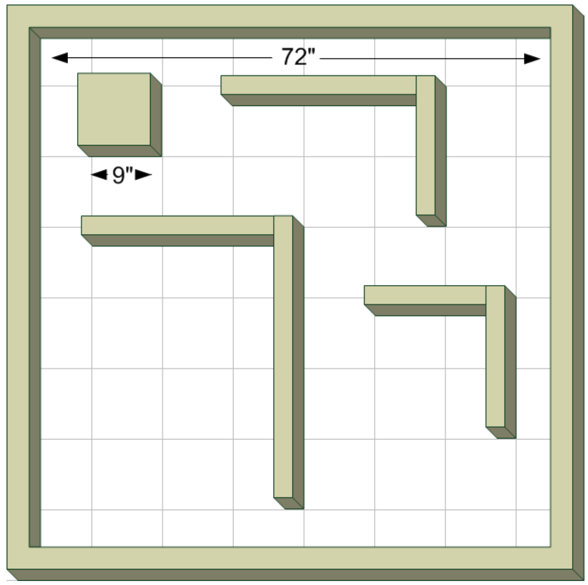
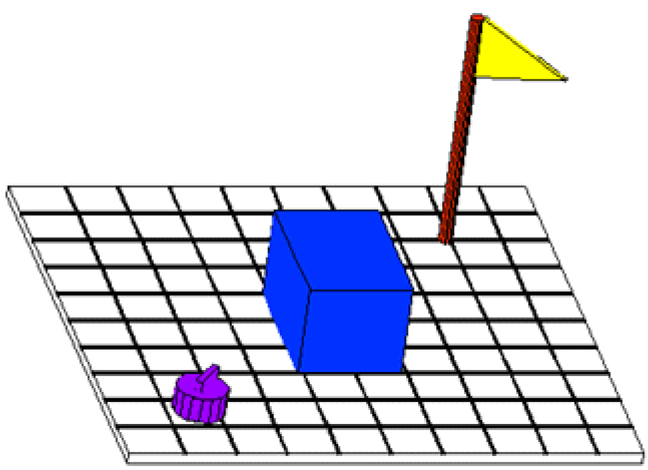


Figure 1: Test Arena

2. The configuration space will be an occupancy grid divided into 9" x 9" squares, where free space is represented by 0's and occupied space by 99's. You should devise a scheme to represent the robot's start position and goal position. Your code should be flexible such that these values can be specified at run time (see Figure 2).



0	0	0	0	0	0	0	0
0	0	0	0	0	0	**0**	0
0	0	0	0	0	0	0	0
0	0	0	**99**	**99**	**99**	0	0
0	0	0	**99**	**99**	**99**	0	0
0	0	0	**99**	**99**	**99**	0	0
0	0	0	0	0	0	0	0
0	**S**	0	0	0	0	0	0

a. Real world b. Configuration Space (8 x 8 matrix)

Figure 2: Test Arena

3. The wavefront is created by starting at the destination and creating eight connected neighbors back to the start point. The robot then follows the numbers in the reverse order to arrive at the goal point (see Figure 3). The goal is for the robot to always move such that the steps to the goal position are reduced. Note that you may need to grow the obstacles by the robot's width to avoid clipping them.

7	6	5	4	3	2	1	1	1
7	6	5	4	3	2	1	**0**	1
7	6	5	4	3	2	1	1	1
7	6	5	**99**	**99**	**99**	2	2	2
7	6	6	**99**	**99**	**99**	3	3	3
7	7	7	**99**	**99**	**99**	4	4	4
8	8	8	7	6	5	5	5	5
9	**9**	8	7	6	6	6	6	6
10	9	8	7	7	7	7	7	7

Figure 3: Test Arena Wavefront

4. During the demonstration, you will be given the map as a 8 x 8 array of 0's and 99's in .txt file that represents free space and obstacles. You will be given the robot's start position at the beginning of the demonstration. Your program should open the .txt file, read the world map as an array, run the wavefront algorithm and plan the path to move the robot from the start position to the goal. You should then place your robot at the start position and press start and it should move to the goal point. You will be graded on how well your algorithm works; the efficiency of the path chosen by the robot, the ability of the robot to reach the goal point while also avoiding obstacles. *(Note that the robot's center of rotation is between its wheels not the center of the chassis so you should offset the robot in the starting cell so that the robot's center of rotation is at the center of the cell. Another technique to prevent the robot from hitting walls and obstacles is to select the path that maximizes the distance between walls and obstacles or the center line.)*

Demonstration:

The demonstration of the program for lab 7 will include the robot navigating from the start point to the goal point using wavefront expansion on the metric map (occupancy grid). The list of robot commands and generated wavefront should be shown on the GUI to make it evident the state that the robot is in.

Program:

The program should be properly commented and modular with each new behavior representing a new function call. The design of the architecture should be evident from the program layout. You should use the GUI, keypad, LCD and speech module as needed to illustrate robot state, input and output data.

Memo:

The following list provides the basic guidelines for writing a technical memorandum.

- ✓ Format
  - Begins with Date, To , From, Subject
  - Font no larger than 12 point font
  - Spacing no larger than double space
  - Written as a paragraph not bulleted list
  - No longer than three pages of text
- ✓ Writing
  - Memo is organized in a logical order
  - Writing is direct, concise and to the point
  - Written in first person from lab partners
  - Correct grammar, no spelling errors
- ✓ Content
  - Starts with a statement of purpose
  - Discusses the strategy or pseudocode for implementing the robot paths (may include a flow chart)
  - Discusses the tests and methods performed
  - States the results including error analysis
  - Shows data tables with error analysis and required plots or graphs
  - Answers all questions posed in the lab procedure
  - Clear statement of conclusions

Questions to Answer in the Memo:

1. What was the strategy for implementing the wavefront algorithm?
2. Were there any points any 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 of algorithm did you use to selection 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 that robot's navigation or wavefront algorithm?

Grading Rubric:

The lab is worth a total of 30 points and is graded by the following rubric.

Points	**Demonstration**	**Code**	**Memo**
10	Excellent work, the robot performs exactly as required	Properly commented, easy to follow with modular components	Follows all guidelines and answers all questions posed
7.5	Performs most of the functionality with minor failures	Partial comments and/or not modular with objects	Does not answer some questions and/or has spelling, grammatical, content errors
5	Performs some of the functionality but with major failures or parts missing	No comments, not modular, not easy to follow	Multiple grammatical, format, content, spelling errors, questions not answered
0	Meets none of the design specifications or not submitted	Not submitted	Not submitted

Submission Requirements:

You must submit the lab memo and code by midnight on *Thursday*. You must also submit a memo for Lab 6 by midnight on *Thursday*.