Designing a Better Vacuum

Your team is doing consulting work for an electronics manufacturer that is designing a new line of automatic vacuum robots. These robots are unique because they contain sensing capabilities that can map out the floor plan of a room before they begin cleaning. Your team has been charged with developing the algorithms that will control the different models of robotic vacuums.

Specifications

- The manufacturer plans to develop three models of robotic vacuums:
  - The economy robot. This robot should clean 100% of the floor in a floor plan, even if the surface has been detected to be clean. The battery life of the economy robot is poor.
  - The regular robot. This robot should clean 100% of the floor in a floor plan, even if the surface has been detected to be clean. The battery life of the regular robot is average.
  - The high-end "intelligent" robot. This robot should only clean the dirty parts of the floor. The battery life of the high-end robot is poor.
- Your algorithms should be written in MATLAB code. In addition to the code, the manufacturer requests pseudocode describing the algorithms created saved as Adobe PDFs.
- The "sensing capabilities" of the robot will generate a matrix. You may assume that the manufacturer has a separate interface to generate the matrix, so you will not need to develop the sensor—instead, you will write your code assuming that the matrix is provided by some separate mechanism. Test matrices are provided by the manufacturer. Additionally, a blank floor plan is provided for creating your own custom floor plan.
- Before the manufacturer will develop prototypes, they want a presentation describing your designs and a brief performance analysis of each robot.

Schedule and Deliverables

PseudoCode: 10%

Create pseudocode for each image recognition function.

Save each pseudocoded function as a separate file with the following naming scheme:

- The economy model (saved as economy_TEAM.pdf)
- The regular model (saved as regular_TEAM.pdf)
- The high end model (saved as highend_TEAM.pdf)

where TEAM is your team identifier. You may create additional functions if you discover a feature that will be common to all three models—those functions should follow the same file naming scheme.

The team captain will turn in the pseudocode to the captain's Blackboard Project Pseudocode assignment.

Source Code: 89%

Each program/function must contain:

- A standard header similar to the one in Chapter 7 of the textbook, plus:
  - Team number
  - First and last names of each team member
  - Documentation of all variables used in the code
- Comments throughout the code

You will write 3 image processing functions. These should show the robot moving across a floor plan so that it's clear what is happening. It should move quickly enough to show a floor cleaned in less than a minute, but slowly enough so that its movement is clear and trips to the recharger are clearly shown. Save each image processing function as a separate file with the following naming scheme:

- The economy model (saved as economy_TEAM.m)
- The regular model (saved as regular_TEAM.m)
- The high end model (saved as highend_TEAM.m)

In addition, you should create a main program that will allow the user to select (using menus or something similar) the file containing the floor plan and the model. The program should prompt for the filename containing the floor plan, then prompt for the desired sheet name in that file. Save this program as VacuumProject_TEAM.m, where TEAM is your team identifier.

In addition to the requirements defined above, each robot should display a summary (after cleaning a floor plan), including:

- The ratio of clean floor to dirty floor. For example "36% of the floor was covered with dirt."
- The area of the floor plan. You may assume that each "pixel" represents a 1 ft x 1 ft tile. "Floor plan surface area: 1200 square feet
- The amount of time it took the robot to clean the floor. You may assume that it takes 5 seconds to clean a single dirty "pixel" on the floor. It takes 1 second to traverse a pixel without cleaning it.
- The number of times the robot had to recharge. "The robot had to recharge 5 times."
    - A robot with "poor" battery life will be able to clean 250 square feet before recharging.
    - A robot with "average" battery life will be able to clean 350 square feet before recharging.
    - You may assume that it takes every robot 60 seconds to recharge. When the robot's battery is empty, it will travel back to the charge station at a rate of 1 pixel per second. After it is fully charged, it will return to where it left off at a rate of 1 pixel per second (without cleaning on the return path.)
    - If the robot isn't cleaning, you may assume that no battery is being used. You may move across the floor at an angle. For angled movements, it is not necessary for the movement to be made up of horizontal and vertical movements. You may move at any diagonal angle.

Your source code will be evaluated based on:

- **Documentation (30%)** - all code must include adequate header information and comments throughout the source code
- **Reusability (25%)** - all code repeated frequently throughout your project should appear in well documented functions
- **Specification (25%)** - all code should meet the project requirements outlined in the project description. Note that your code will be tested on secret floor plans. Your code will be evaluated not only on how well it does on the four floor plans that I have distributed, but also on my secret floor plans.
- **Efficiency (20%)** - No extraneous calls to unnecessary functions or other poor software design decisions.

The team captain will turn in the Source Code as a .zip file to the captain's Blackboard Project Source Code assignment.

Peer Evaluation: 1%

You get 1% by turning in your peer evaluation of yourself and your teammates. Furthermore, the sum of your Pseudocode, Working Code and Peer Evaluation grades is multiplied by the average peer evaluation you receive from members of your group. For example, if your Pseudocode grade is 8/10, your Working Code grade is 80/89, you turn in your Peer Evaluation and your average Peer Evaluation is 80/100, your grade is (8+80+1)*80/100 = 71.2. Everyone will turn in a Peer Evaluation to their own Blackboard Project Peer Evaluation assignment. The rating is just a number from 0 to 100 for yourself and each of your teammates.

Floor Plan Matrix

The manufacturer has developed a sensing mechanism that will generate a 30 x 25 matrix of values that represent different values, as defined below. The algorithms developed should use this matrix as the input to clean the floor.

Floor Plan Matrix Values:

- 0 A wall, couch, or any other surface where the vacuum cannot travel.
- 4 Clean floor
- 3 Dirty floor
- 2 Charging station
- 1 The current location of the robot in the image

You may assume that the matrix will always be bordered by a wall and the charging station is always in the same location.

Creating Images in MATLAB

Assume we read in the first floor plan to a matrix, A. In an image, we will refer to the “values” in a matrix as pixels.

To plot the matrix A as an image:

```
map=[
    0 0 0;                        %black, inaccessible
    1 0 0;                        %red, current robot location
    0 0 1;                        %blue, charging station
    127/255 50/255 50/255  ;      %brown, dirty floor
    1 1 1;                        %white, clean floor
    ];

image(A) % display the contents of matrix A
colormap(map) % Set the colormap to map
axis('image') % resize the plot in the figure to the aspect ratio of
% the matrix
axis off % turn off the axis labels
```

The above segment of MATLAB code will create the image shown below.

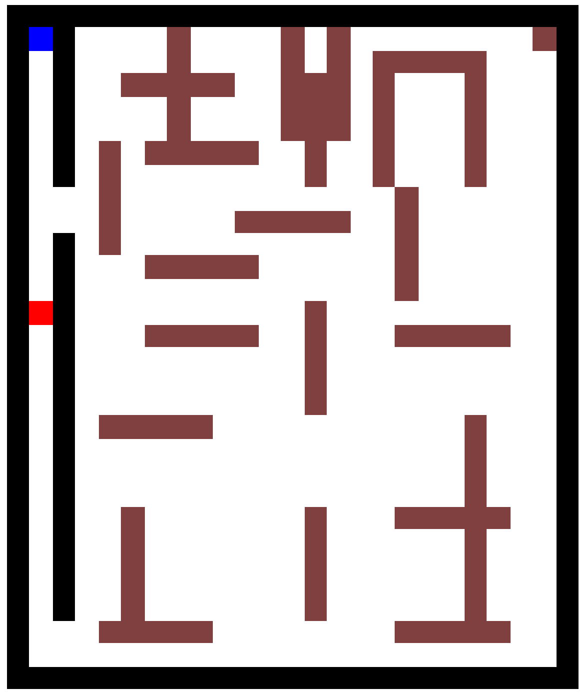

Image of a floor plan showing the walls, dirt, floor, and charging station (top left) . The robot is in red.