ECE 172A
Ruomei Ye
A99074214

Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.
By including this in my report, I agree to abide by the Academic Integrity Policy mentioned above.

Problem1
(i).

```
A=[2 59 2 5;41 11 0 4;18 2 3 9;6 23 27 10;5 8 5 1];
B=[0 1 0 1;0 1 1 1;0 0 0 1;1 1 0 1;0 1 0 0];

A =

     2    59     2     5
    41    11     0     4
    18     2     3     9
     6    23    27    10
     5     8     5     1


B =

     0     1     0     1
     0     1     1     1
     0     0     0     1
     1     1     0     1
     0     1     0     0
```

(ii). C=A.*B

```
C =

     0    59     0     5
     0    11     0     4
     0     0     0     9
     6    23     0    10
     0     8     0     0

(iii).inner=dot(C(2,:),C(5,:))

inner =

    88

(iv).
min_value=min(C(:))
max_value=max(C(:))


min_value =

     0


max_value =

    59
[row,col]=find(C==min_value)
[row,col]=find(C==max_value)

row =

     1
     2
     3
     5
     3
     1
     2
     3
     4
     5
     5


col =
     1
     1
     1
     1
     2
     3
     3
     3
     3
     3
     4


row =

     1


col =

     2
```

Therefore the min value is 0, at indices (1,1) (2,1)(3,1)(5,1)(3,2)(1,3)(2,3)(3,3)(4,3)(5,3)(5,4)
The max value is 59 at (1,2)

(v).

```
D=bsxfun(@minus,C,C(1,:))

D =

     0     0     0     0
     0   -48     0    -1
     0   -59     0     4
     6   -36     0     5
     0   -51     0    -5
```

(vi).E=bsxfun(@minus,D,D(1,:))

```
E =

     0     0     0     0
     0   -48     0    -1
```

```
    0  -59    0    4
    6  -36    0    5
    0  -51    0   -5
```

Pro2

(i).

- What does loc keep track of?

It keeps track of the current location of the robot.

- In what direction should the robot move based on line 14 (adding [1, 0])?

South

- Add a new object at [4, 7]. Explain what happens and why.

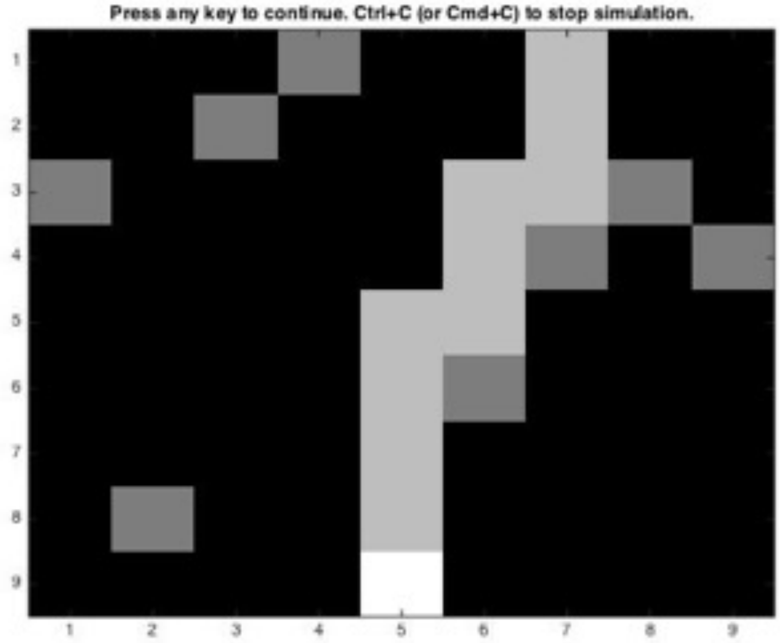
It will stop at the pixel in front of the object at [4,7].

- If the robot were implemented based on the current state of the simulation, briefly explain why this robot would not be considered an intelligent system.

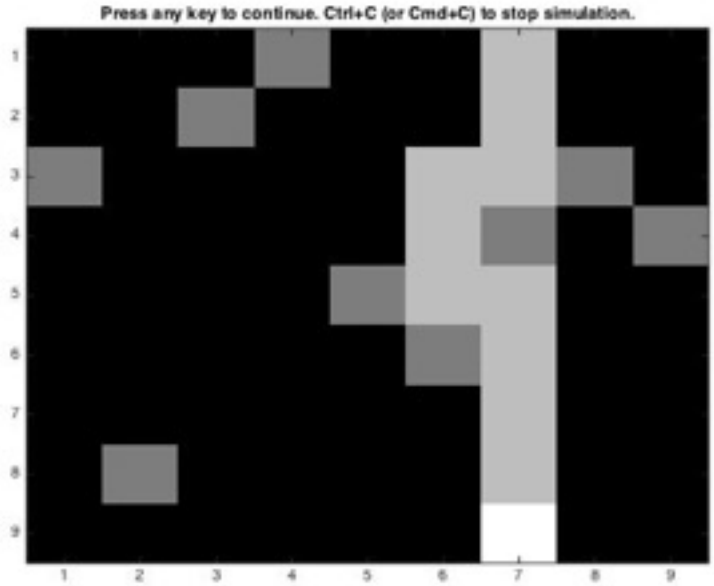
The system is not able to avoid the object when the object is in its moving direction.

(ii).

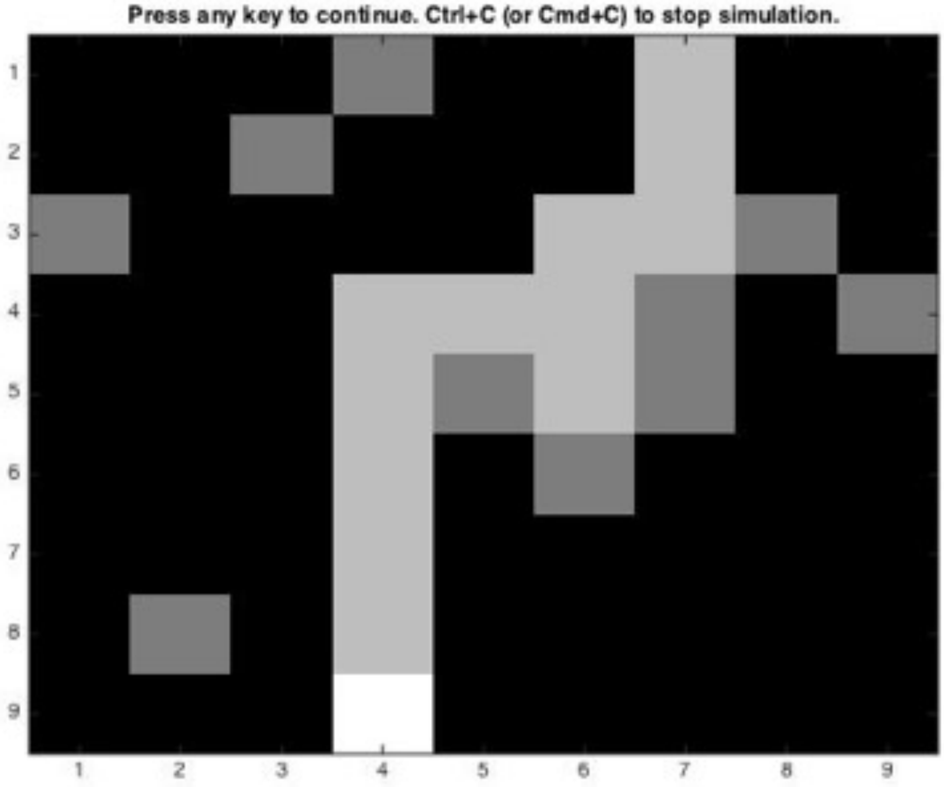
After using the if statement, the robot will not stop when confronting the object; instead, it knows to change its path to avoid the object and continue to move in its original direction.



(iii).


Press any key to continue. Ctrl+C (or Cmd+C) to stop simulation.

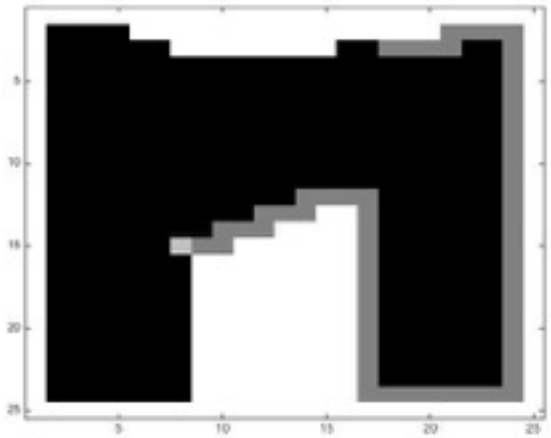
(iv).


Press any key to continue. Ctrl+C (or Cmd+C) to stop simulation.

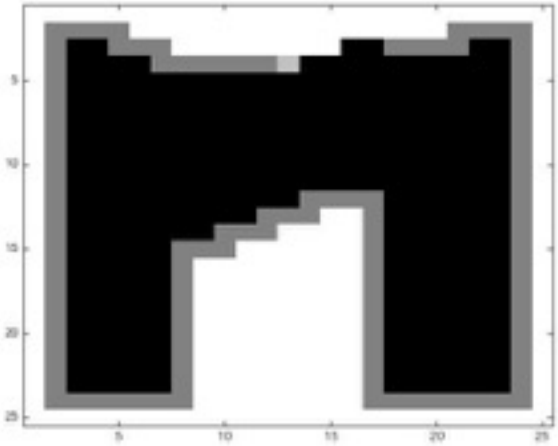
Pro3

(i). clockwise:

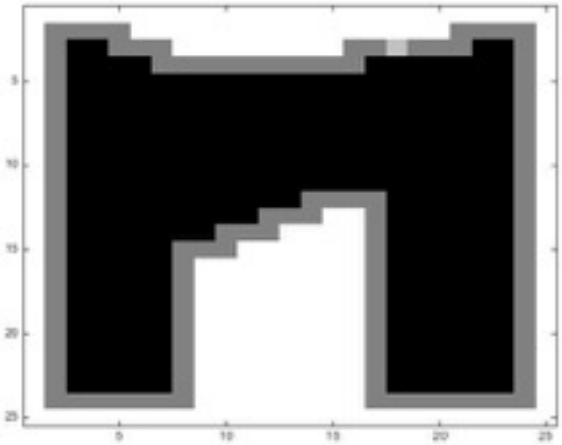
50%:



90%:

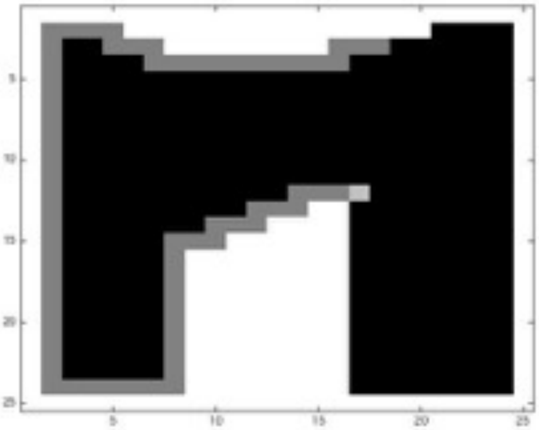


100%:

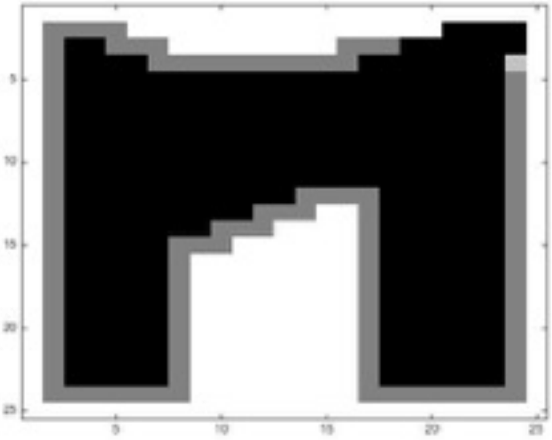


(ii). counterclockwise:

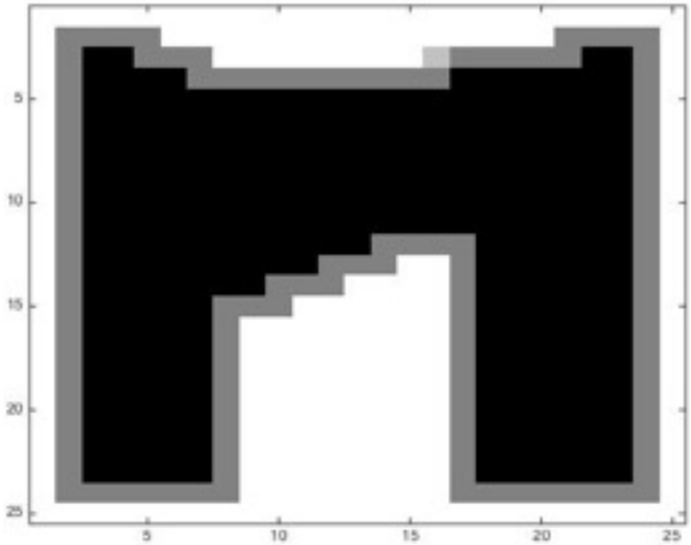
50%:



90%:



100%:



Problem 4

The intelligent robot I am interested in is a human-type robot named RIBA (Robot for Interactive Body Assistance). Its main function is to assist caregivers to move and lift patients between the wheelchair and bed in hospital, and it is also able to help patients during the rehabilitation training. The RIBA achieve such movement by using its human-like arms. The robot was designed by a Japanese team in 2010.

Since the main purpose of RIBA is to transfer humans, the most essential part of it is the tactile sensors. It has sensor points on its upper arm, forearm, hand and shoulder pad to sense the position of the patient and ensure the human being feels comfortable when being lifted by its arms. Also, since the robot was designed to interact with humans, it has audio sensors, which are microphones, to receive commands from caregivers and make sounds when necessary. Lastly, since the RIBA will move the patient on to a bed or wheelchair, it has vision sensors, which are the cameras to detect the position of the bed and the wheelchair.

The robot is powered by a DC motor inside it, and is controlled by a method named "tactile guidance". It is not controlled by motion capture or remote controller because the directly touch and gesture will be more useful for caregivers to utilize the robot. Lifting human beings in a comfortable way is not very easy to achieve because the process requires very accurate and sensitive sensors and requires the robot to adjust its force, position and gestures frequently. To solve such problem, the RIBA is directly guide by the caregivers — in other words, when the caregivers believe some adjustments are needed during the transferring process, they can directly touch the arms of the RIBA and instruct robot to adjust its arms.Therefore, it is very easy and direct for caregivers to make commands and allows the caregivers to make adjustment very frequently by a slight gesture.For instance, when the caregivers would like the robot to set a person on bed, they can simply make a sliding motion on the RIBA's arm and the robot will process the instruction when detecting such physical touching. It is controlled by voice commands as well. The robot has three different modes to accommodate several situations, and humans can change its mode by voice control. The three modes includes cart control mode, posture-forming mode and motion-adjusting mode. These methods are very necessary to customize its service for every individual patient. For example, for patients with different weight and height, the robot needs to use the motion-adjusting mode to adjust its arms' distance and its speed.

The application of such robot, as mentioned at beginning, is in domain of nursing and medical care. I think such intelligent robot is necessary in such area. First, as most caregivers in hospital are females, a patient might be too heavy for nurses to lift and move between the bed and wheelchairs. Also, such relocation happens very frequently in the caregivers' daily work, and the RIBA will solve such difficulty. Moreover, the tactile guidance allows the caregivers to easily make slight adjustment during the moving process, instead of using a remote control to carefully control the machine. Meanwhile, RIBA could provide a more comfortable service than other human-type machines during transferring a patient as it is designed for this specific task — the RIBA robot could adjust the degrees and slope of its bending arms, and the distance between two arms as well as its speed when moving different patients. Other machines might not be able to make such very accurate and specific adjustments.

However, since such robot is mainly controlled by tactile sensors, I suppose it will face some difficulties when developing its vision subsystem. First, since it only has two cameras, its vision sight is not very wide and it cannot detect every corner around it. Therefore, it might miss some potential obstacles. Moreover, I think the main purpose of its vision sensor is to detect some major objects, such as the patient, bed and wheelchair. It might be challenging for it to distinguish similar objects in some situations and detect potential dangerous objects. For instance, if there are some small objects on the bed, the robot will still just put the patient on it.

Apart from the limitation of its vision sensor, another technical issue I am concerned with is that the RIBA cannot independently complete the entire process of transferring without the assistance of caregivers. This is because it is controlled by the physical contact and requires the caregivers to adjust it frequently. To popularize the use of such robot will require the improvement of its ability to work independently and to learn from the instructions better, such that it will complete the whole process with less help from humans and is able to learn and memorize the adjustments it needs to make for different types of patients. For example, it should be improved to learn to decrease the arm distance when moving a kid even without the instruction from caregivers.

The ethical issues it might have is that if it is improved to have more functions, and could perfectly moving and lifting the patients in hospital, it might cause the loss of job for human beings. Also, if some accidents happen. it might be hard to determine if it is because of the wrong instruction from the caregivers or because of the technical issues of the robot itself.

Works Cited

Toshiharu Mukai, Shinya Hirano, Hiromichi Nakashima, Yo Kato, Yuki Sakaida, Shijie Guo and Shigeyuki Hosoe. “Development of a Nursing-Care Assistant Robot RIBA That Can Lift a Human in Its Arms” . Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on.

Appendix:

Problem1:

```
(i). A=[2 59 2 5;41 11 0 4;18 2 3 9;6 23 27 10;5 8 5 1];

B=[0 1 0 1;0 1 1 1;0 0 0 1;1 1 0 1;0 1 0 0];


(ii). C=A.*B

(iii). inner=dot(C(2,:),C(5,:))

(iv).
```

```
min_value=min(C(:))

max_value=max(C(:))

[row,col]=find(C==min_value)

[row,col]=find(C==max_value)
```

(v).

```
D=bsxfun(@minus,C,C(1,:))
```

(vi).

```
E=bsxfun(@minus,D,D(1,:))
```

Problem 2

```
(iii).  %   START
 if detectObject(loc, obj, 'S')
         nextStep = loc(end,:) + [0 -1];

        if detectObject(loc,obj,'W')
            nextStep = loc(end,:)+ [0 1];
        end
     end
   %  STOP
```

(iv).

```
  %   START
    if haveIBeenHereBefore(loc,nextStep)
        nextStep = loc(end,:) + [0 -1];
    end

     if detectObject(loc, obj, 'S')
         nextStep = loc(end,:) + [0 -1];

        if detectObject(loc,obj,'W')
            nextStep = loc(end,:)+ [0 1];

            if detectObject(loc,obj,'E')
                nextStep = loc(end,:)+ [-1 0];
             %   if haveIBeenHereBefore(loc,nextStep)
            end
        end
     end
   %  STOP
```

Problem3:

```
function [ newPos ] = get_new_pos( curPos, sensorInput, dir )
% sensorInput is a 3x3 matrix and follows the following convention.
% sensorInput = [   a   b   c   ;
%                   d   x   e   ;
%                   f   g   h   ];
% 'x' is the location of the bot.
% a to h are either '0' or '1' depending on the presence of obstacle in
% those blocks. For example take the image shown in Q2. At this location
% sensorInput = [1 1 1;
%                0 0 0;
%                0 0 0];
    newPos = curPos;
    if dir == 0
        if sensorInput(1,2) == 1
            newPos(2) = newPos(2) + 1; % Move Right
            %newPos(1) = newPos(1) + 1; % Move Down
        end

        if sensorInput(1,2) == 0 && sensorInput(1,1) == 1
            newPos(1) = newPos(1) - 1; %  up
        end

        if sensorInput(2,3) == 1 && sensorInput(1,2) == 1 && sensorInput(3,2)==0
            newPos(1) = newPos(1) + 1;
            newPos(2) = newPos(2) -1; %down corner
        end

        if sensorInput(3,2) == 0 && sensorInput(2,3) == 1 && sensorInput(1,2)==0
            newPos(1) = newPos(1) + 1;  % down
        end
        if  sensorInput(3,2) == 1 && sensorInput(2,1)==0
            newPos(2) = newPos(2) - 1;  %left
        end

        if  sensorInput(2,1) == 1 && sensorInput(1,2)==0 && sensorInput(1,1)==0 &&
sensorInput(3,2)==0
            newPos(1) = newPos(1) - 1;  %up
        end
        if  sensorInput(3,1) == 1 && sensorInput(2,1)==0 && sensorInput(3,2)==0
            newPos(2) = newPos(2) - 1;  %left at corner
        end
        if  sensorInput(3,2) == 0 && sensorInput(3,3)==1 && sensorInput(2,3)==0
            newPos(1) = newPos(1) + 1;
        end
        if  sensorInput(1,3) == 1 && sensorInput(2,3)==0 && sensorInput(1,2)==0
            newPos(2) = newPos(2) + 1;
        end


    elseif dir == 1
        if sensorInput(1,2) == 1  || (sensorInput(1,1) == 1 && sensorInput(2,1) == 0)
            newPos(2) = newPos(2) - 1; % Move Left  _|
        end
        if sensorInput(2,1) == 1 &&  sensorInput(1,2) == 1
            newPos(1) = newPos(1) + 1;
            newPos(2) = newPos(2) + 1;    %Move down at corner zuoshang
        end
        if (sensorInput(3,1) == 1 &&  sensorInput(3,2) == 0 && sensorInput(2,1) == 0)
            newPos(1) = newPos(1) + 1;
              %Move down
        end


        if sensorInput(2,1) == 1 && sensorInput(1,2) == 0
            newPos(1) = newPos(1) + 1;  %move down not at corner
        end
        if (sensorInput(3,2) == 1 && sensorInput(2,1) == 1)
            newPos(2) = newPos(2) + 1;
            newPos(1) = newPos(1) - 1;  %move right at |_corner
        end
        if sensorInput(2,3) == 1 && sensorInput(1,2) == 0 && sensorInput(1,3) == 0
            newPos(1) = newPos(1) - 1;

        end
        if sensorInput(2,3) == 0 && sensorInput(3,3) == 1 && sensorInput(3,2) == 0
            newPos(2) = newPos(2) + 1;
        end
        if sensorInput(3,2) ==1 && sensorInput(2,1) == 0
            newPos(2) = newPos(2) +1; %right
        end
        if ((sensorInput(1,3)) == 1 && (sensorInput(1,2) ==0))%||(sensorInput(2,3) ==1 &&
    sensorInput(1,2) == 0 && sensorInput(3,2) ==0 &&sensorInput(1,3)==0)
          newPos(1) = newPos(1) -1;

           %newPos(1) = newPos(1) - 1; % Move Up at |_ or |-corner
        end
        if sensorInput(2,3) ==1 && sensorInput(1,2) == 0 && sensorInput(3,2) ==1
          newPos(1) = newPos(1) -1;
          newPos(1) = newPos(1) +1;
          newPos(2) = newPos(2) -1;

           %newPos(1) = newPos(1) - 1; % Move Up  at _| corner
        end

      end
    end
```