Quiz 10: CS4640 Name ____________________

1. Given the following image, and supposing that the pixels with value 1 constitute a shape of interest, give the set X of points in the shape (in x,y values where the origin is at the upper left corner of the image as used in the text), give the anchor point, P, (use the mean), and the Hough shape model, D. Also, label the x- and y-axes of the image.

```
        y
    1 2 3 4 5
  +----------->
 1| 0 0 0 0 0
 2| 0 1 0 1 0
x 3| 0 0 0 0 0
 4| 0 1 0 1 0
 5| 0 0 0 0 0
  V
```

```
X = {[2;2], [2;4], [4;2], [4;4]}

P = [3;3]

D = {[1;1], [-1,1], [1;-1], [-1,-1]}
```

2. Given the Hough shape model:

```
D = {[0;0], [1;1], [1;0], [1;-1]}
```

and the image:

```
   image                        Hough shape accumulator
0 0 0 0 0 0 0 0                  0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0                  0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0                  1 2 2 1 0 0 0 0
0 0 0 0 0 1 0 0                  0 0 0 0 0 1 0 0
0 0 0 0 1 1 1 0                  0 0 0 0 2 2 2 0
0 1 1 0 0 0 0 0                  0 1 1 1 2 3 2 1
0 0 1 0 0 0 0 0                  1 2 2 1 0 0 0 0
0 0 0 0 0 0 0 0                  0 0 0 0 0 0 0 0
```

determine the Hough shape accumulator array values (show as an image of appropriate size and values from 0 to whatever max is achieved).

3. Suppose we want a Hough line detector that finds the a,b,c parameters for the equation $ax + by + c = 0$, where $[a; b]$ is the unit normal to the line, and c is the minimum perpendicular distance to the line from the origin.

3a. Explain how this could be most efficiently set up (i.e., minimum number of parameters), and explain how the accumulator works, and what a reasonable set of bin values would be.

Run exactly like the (e.g.) Matlab Hough transform or as indicated in the text and class. Then use the resulting θ and ρ to get a, b and c:

$$a = cos(\theta)$$

$$b = sin(\theta)$$

$$c = -\rho$$

3b. Describe in detail how your method from 3a would work on the following image (give the accumulator values as well as the indexing values; some useful values: $cos(\pi/4) = sin(\pi/4) = 0.7$ and $cos(3\pi/4) = -sin(\pi/4) = -0.7$).

```
  image                           H

0 0 1 0 0 0          5.6 |    1
0 0 1 0 0 0          5   | 2  1
0 0 1 0 0 0          4.2 |    1
0 0 0 0 0 0          3.5 |    1
0 1 1 0 0 0   rho    3   | 1  1  4     1
0 0 0 0 0 0          2   | 1
                     1.4 |       1     1
                     1   | 1
                     0   |             1
                    -0.7 |             1
                    -1.4 |             1
                         +-------------------
                           0  pi  pi  3*pi
                              --  --  ----
                              4   2    4

                                theta
```

4. Suppose we want to set up a general Hough circle detector that works similarly to the Hough general line detector. Explain how that would be set up (i.e., give the equation it is based on, and how the accumulator would be set up).

Use implicit equation:

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

where $[x_c, y_c]^T$ is the center and r is the radius. Then use a 3D accumulator where the center can be any pixel in the image (assuming the center is in the image), and the radius can be from r_{min} to r_{max}. Then let the x_0 values be from 1 to M, and the y_0 be from 1 to N. Calculate the r value as:

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

and increment $H(x_0, y_0, r)$.

5. Give the Matlab code for the following header.

```
function line_im = CS4640_line_im(M,N,a,b,c)
% CS4640_ine_im - sets to 1 all image pixels near line ax + by + c = 0
% On input:
%     M (int): number of rows in image
%     N (int): number of columns in image
%     a (float): coefficient for x in equation: ax + by + c = 0
%     b (float): coefficient for y in equation: ax + by + c = 0
%     c (float): constant coefficient in equation: ax + by + c = 0
% On output:
%     line_im (MxN array): image with line ax + by + c = 0 pixels on
% Call:
%     l1 = CS4640_line_im(7,7,0,1,-3);
% Author:
%     Quiz Taker
%     UU
%     Spring 2018
%

ZERO_THRESH = sqrt(2) + 0.01;

line_im = zeros(M,N);

for x = 1:M
    for y = 1:N
        if abs(a*x+b*y+c)<ZERO_THRESH
            line_im(x,y) = 1;
        end
    end
end
```