Table of Contents

Q1.1

```
%The location for each pixel
nX = 200; %number of columns
nY = 200; %number of rows
xj = 1:nX; %x location of each column
yi = 1:nY; %y location of each row

[xij,yij] = meshgrid(xj,yi); %x,y location of each pixel

%define an image of an annulus
cx = 75; % x component of center
cy = 75; % y component of center
r1 = 30; % inner radius in pixels
r2 = 45; %outer radius in pixels
% define the image using binary equations
I = ((xij - cx).^2 + (yij-cy).^2 <= r2^2 )- ((xij-cx).^2 + (yij-cy).^2
 < r1^2);

%display it
figure;
grid on;
imagesc(I);
axis image;
title('I(x)');
set(gca,'ydir','normal'); % put origin at bottom left
```

Q1.2

```
%set the coordinates of landmarks in I
X = [110 75;95 105; 55 105;37.5 75;55 42.5;95 42.5];
%set the coordinates of landmarks in J
Y = [180 120; 150 170; 90 170;60 120;90 70;150 70];
```

Q1.3

```
syms s
```

```
eqn=0;
%find scale factor
for i = 1:max(size(X))
    for j = 1:min(size(X))

        eqn=eqn+diff((abs((s*X(i,j)-Y(i,j))))^2,s);
    end

end

scaleFactor = double(solve(eqn==0,s));
```

Q1.4

```
%transformed landmarks by scale factor
sX = scaleFactor.*X;

hold on;
scatter(X(:,1),X(:,2),'c');
scatter(sX(:,1),sX(:,2),'b');
scatter(Y(:,1),Y(:,2),'r');
```

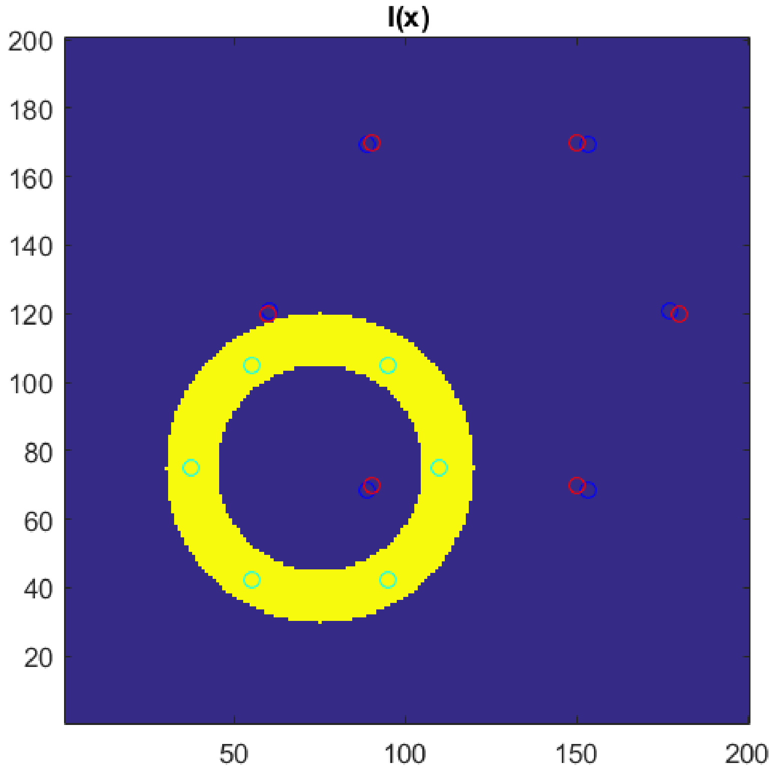
Q1.5

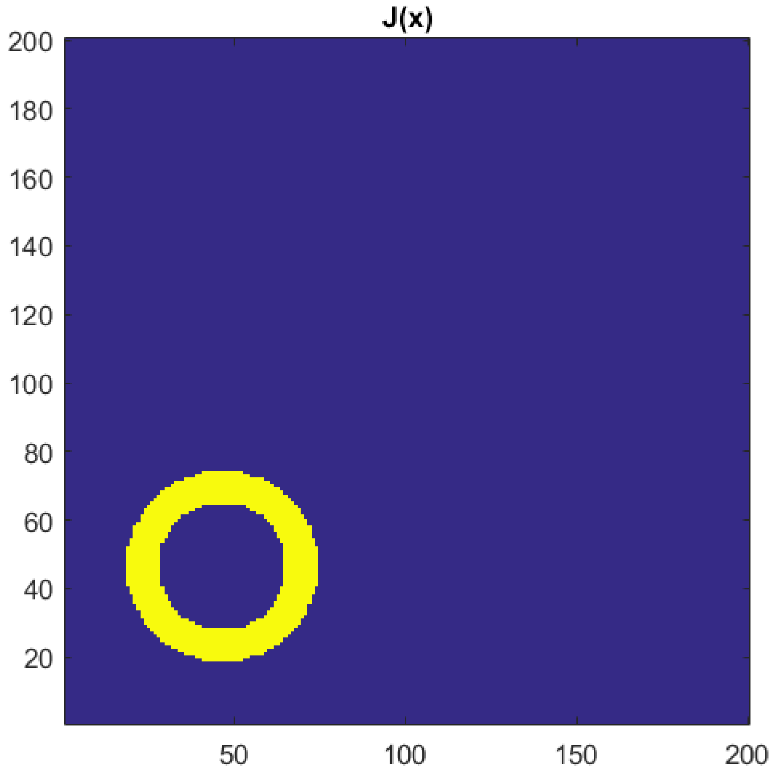
```
%initialize an image of all zeroes
ITransformed = zeros(size(I));

for i = 1:nY %loop through each row
    for j= 1:nX %loop through each column

        %we are looking for the value to assign to Isx(j,i)

        %find the posiiton to look at in the image J
        iLook = i*scaleFactor;
        jLook = j*scaleFactor;

        %round them to the nearest integer
        iLookRound = round(iLook);
        jLookRound = round(jLook);

        %check if we're out of bounds,
        if iLookRound < 1 || iLookRound > nY || jLookRound < 1 ||
 jLookRound > nX
            %if so, fill the image with the value zero
            ITransformed(j,i) = 0;
        else
            %otherwise, assign the value in our image at this point
            ITransformed(j,i) = I(jLookRound,iLookRound);
        end
        %don't forget to index your images by (row,column) and not
 (x,y)!
    end
```

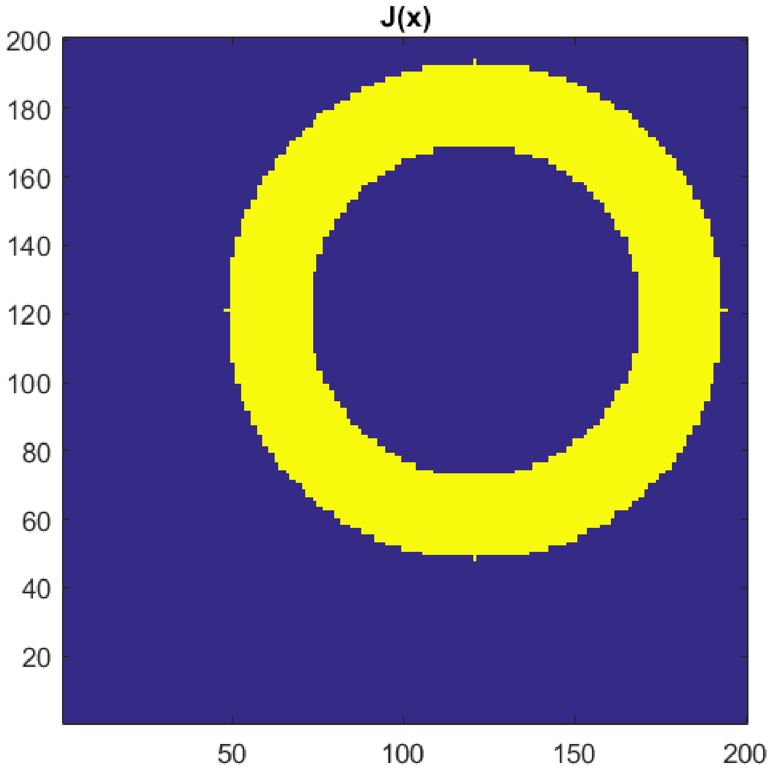
```
end
figure
grid on
imagesc(ITransformed);
axis image;
title('J(x)');
set(gca,'ydir','normal');
```

Q1.6

```
%initialize an image of all zeroes
ITransformed = zeros(size(I));

for i = 1:nY %loop through each row
    for j= 1:nX %loop through each column

        %we are looking for the value to assign to Isx(j,i)

        %find the posiiton to look at in the image J
        iLook = i*(scaleFactor^-1);
        jLook = j*(scaleFactor^-1);

        %round them to the nearest integer
        iLookRound = round(iLook);
        jLookRound = round(jLook);

        %check if we're out of bounds,
        if iLookRound < 1 || iLookRound > nY || jLookRound < 1 ||
 jLookRound > nX
            %if so, fill the image with the value zero
            ITransformed(j,i) = 0;
        else
            %otherwise, assign the value in our image at this point
            ITransformed(j,i) = I(jLookRound,iLookRound);
        end
        %don't forget to index your images by (row,column) and not
 (x,y)!
    end
end
figure
grid on
imagesc(ITransformed);
axis image;
title('J(x)');
set(gca,'ydir','normal');
```


I(x)
200
180
160
140
120
100
80
60
40
20
50
100
150
200


J(x)
200
180
160
140
120
100
80
60
40
20
50
100
150
200



Published with MATLAB® R2015a