EE417

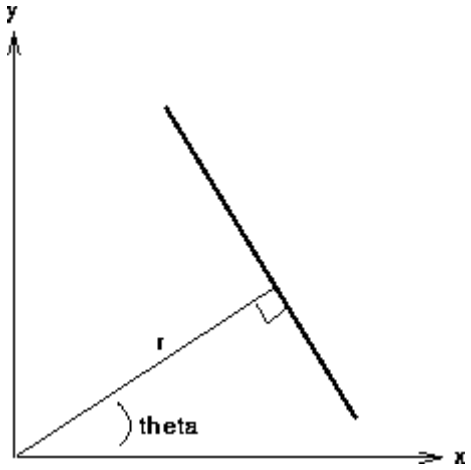
POST-LAB #4 REPORT

1. Line Detection

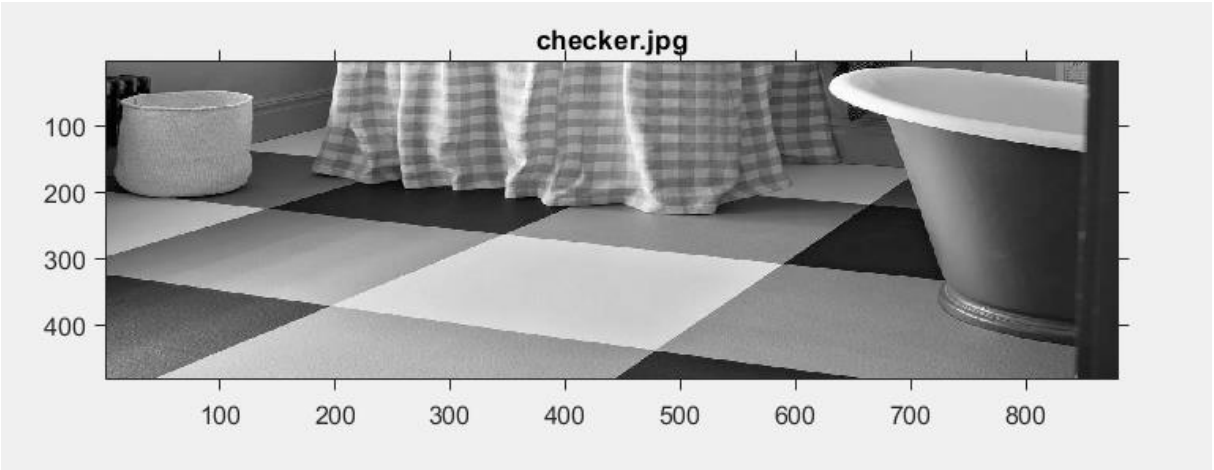
The Hough transform is a technique which can be used to isolate features of a particular shape within an image.

$$x\cos\theta + y\sin\theta = r$$

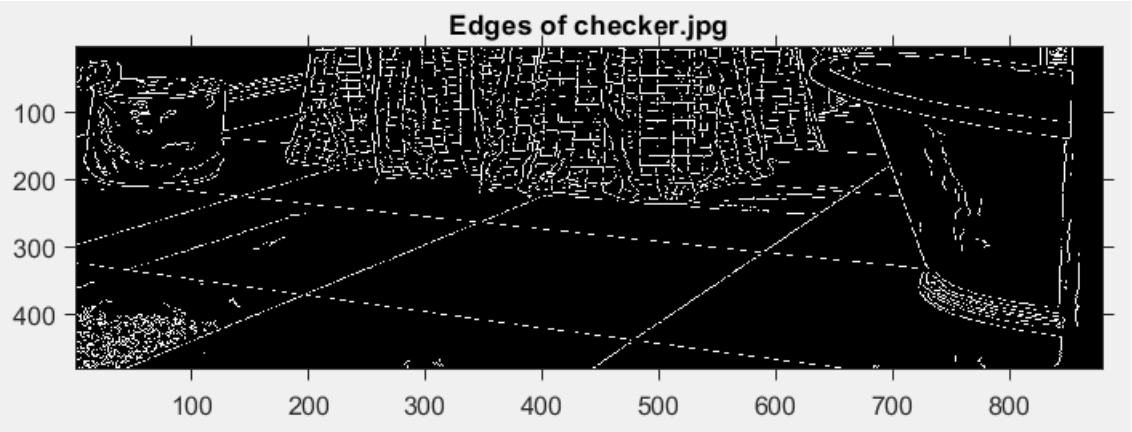
where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis. (See Figure 2.) For any point (x, y) on this line, r and θ are constant.



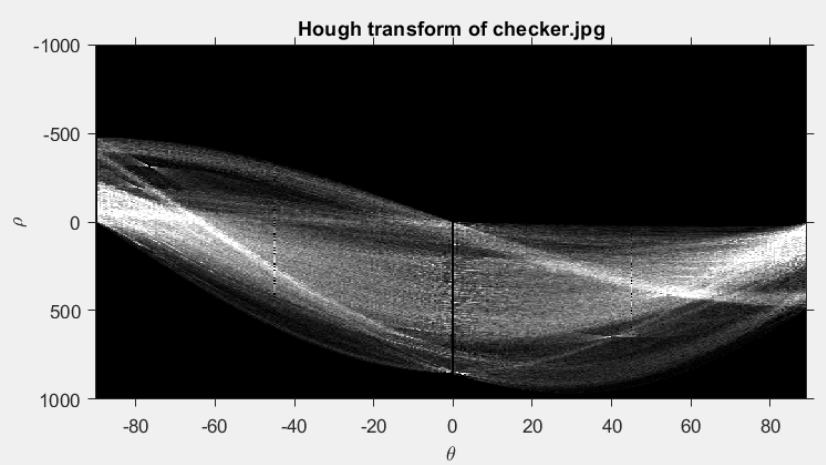
We will use Hough Transform to detect lines in checker.jpg image



Firstly we used Canny Edge Detector to get the edges in order to apply Hough transform.

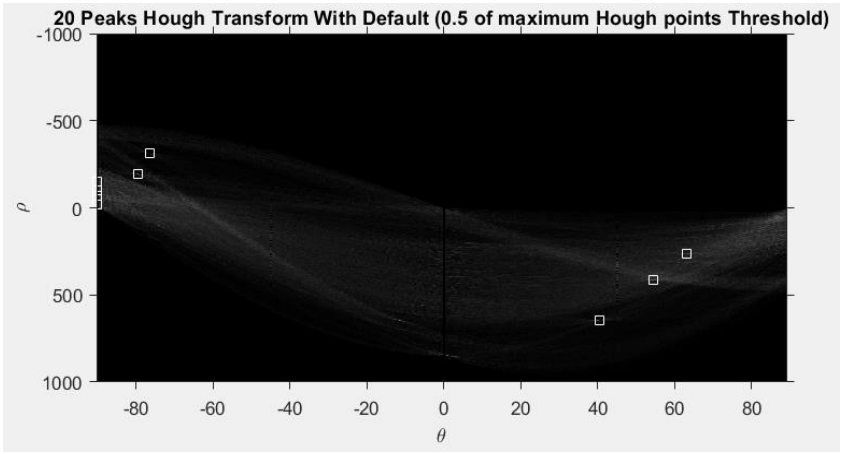


After the edges are detected, we converted the pixels into a ϱ θ plane by use of Hough transform. (Elapsed Time: 0.036140 seconds)

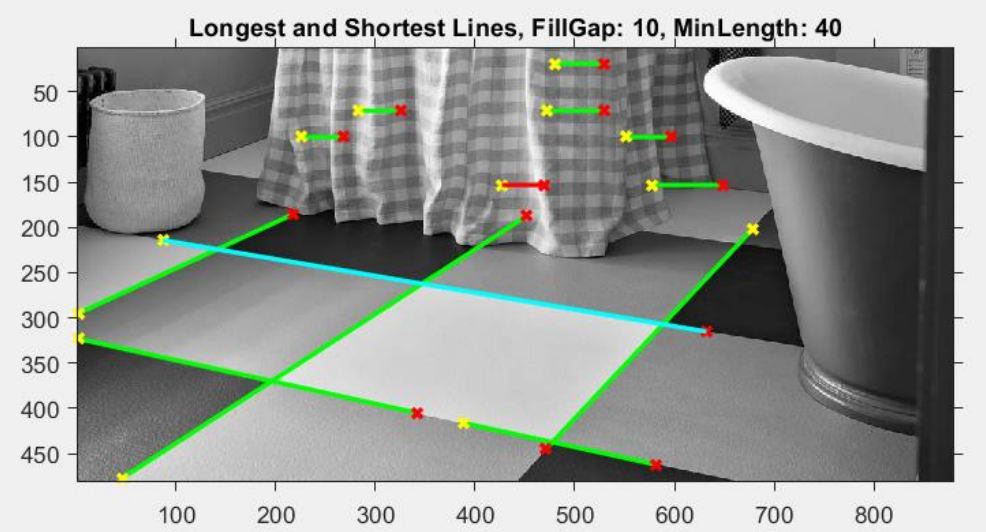


There is a voting system in the nature of Hough transform, we can deduce that non-zero pixels (which are edges) are counted as a vote in the corresponding ϱ and θ value. Brighter points has more counts.

Then, we determined a threshold (as half of the maximum Hough points) in order to eliminate some weak lines that are detected by the Hough transform method. (Elapsed Time: 0.022201 seconds)



Lastly, we get the lines by these "bright spots" which can be seen above. A lot of points "voted" for this spot. And these are the parameters that describe the lines in the original image. (Elapsed Time: 0.006134 seconds)



lab4houghlines.m

```
function [H,T,R,edges,P,lines] = lab4houghlines(img)

    [row,col,ch] = size(img);
    X = zeros(size(img));
    k = 1;

    if(ch==3)
        img = rgb2gray(img);
    end

    edges = edge(img,'canny');

    tic;
    [H,T,R] = hough(edges,'RhoResolution',0.5,'Theta',-
90:0.5:89);
    toc;

    tic;
    P = houghpeaks(H,20,'Threshold',0.5*max(H(:)));
    toc;

    tic;
    lines =
houghlines(edges,T,R,P,'FillGap',10,'MinLength',40);
    toc;
end
```

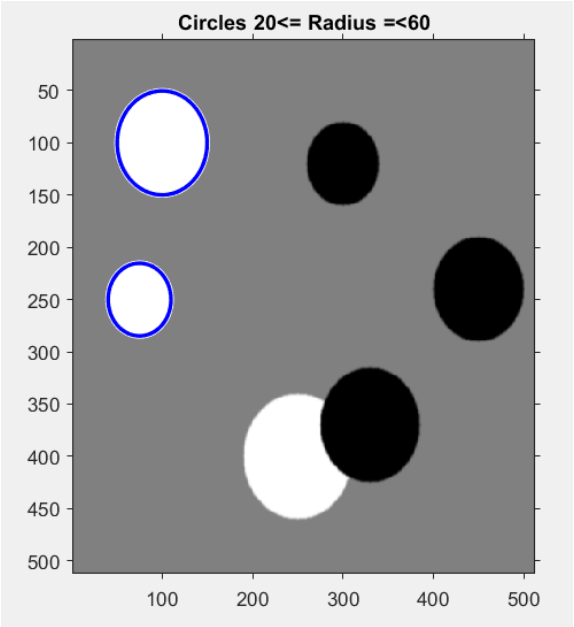
lab4.m (First part)

```
img = imread('checker.jpg');

 [row,col,ch] = size(img);
 if(ch==3)
    img = rgb2gray(img);
 end

[H,T,R,edges,P,lines] = lab4houghlines(img);

subplot(3,2,1);
imshow(img);
title('checker.jpg');
axis on, axis normal, hold on;

subplot(2,2,2);
imshow(imadjust(rescale(H)),'XData',T,'YData',R,...
      'InitialMagnification','fit');
title('Hough transform of checker.jpg');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;

subplot(2,2,3);
imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
title('20 Peaks Hough Transform With Default (0.5 of maximum
Hough points Threshold)');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
plot(T(P(:,2)),R(P(:,1)),'s','color','white');


subplot(2,2,4);
max_len = 0;
min_len = 100000000000;
imshow(img);
title('Longest and Shortest Lines, FillGap: 10, MinLength:
40');
axis on, axis normal, hold on;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
```

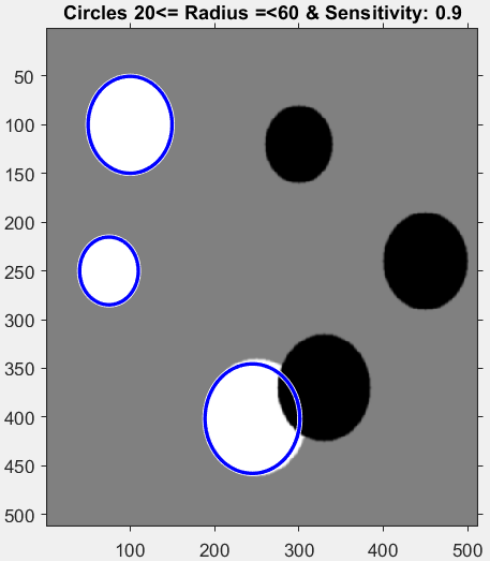
```
    % Plot beginnings and ends of lines
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

    % Determine the endpoints of the longest line segment
    len = norm(lines(k).point1 - lines(k).point2);
    if ( len > max_len)
       max_len = len;
       xy_long = xy;
    end

    % Determine the endpoints of the shortest line segment

    if ( len < min_len)
       min_len = len;
       xy_short = xy;
    end
end
axis on, axis normal, hold on;

 plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','cyan');

plot(xy_short(:,1),xy_short(:,2),'LineWidth',2,'Color','red');

 axis on, axis normal, hold on;
```

2. Circle Detection

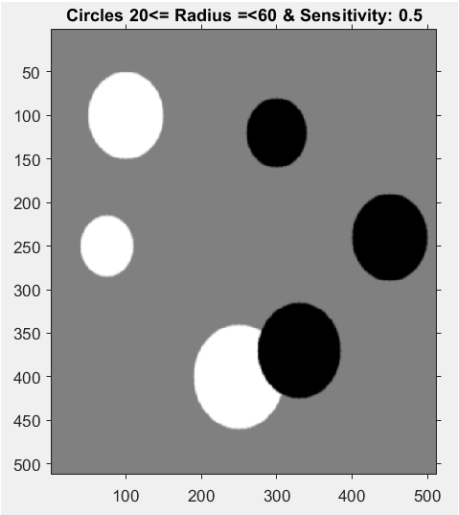
As we mentioned before, Hough transform method can be used to detect for several shapes such as lines, circles. In this section of the lab, we will detect the circular shapes in the circlesBrightDark.png image.
Firstly, we detected the circles that has a radius between 20 and 60 pixels (Elapsed time: 0.213628 seconds):

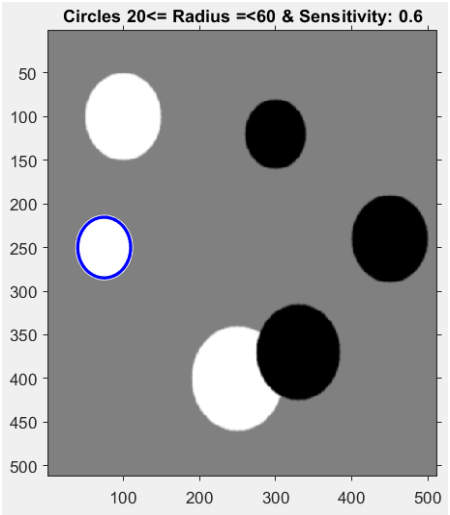


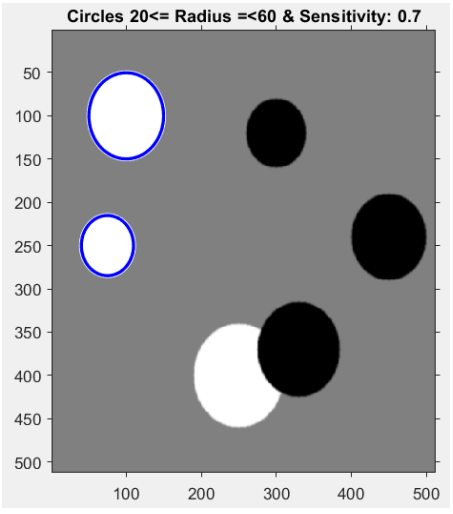
After that we changed the sensitivity factor to 0.9 (default sensitivity factor was 0.85 in MATLAB environment). (Elapsed time: 0.206446 seconds)

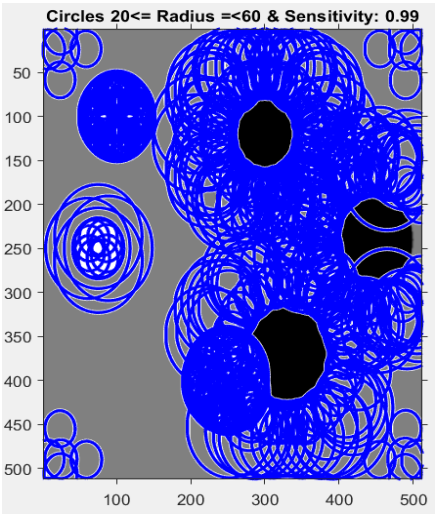


We varied the sensitivity factor of the image to see the effects.



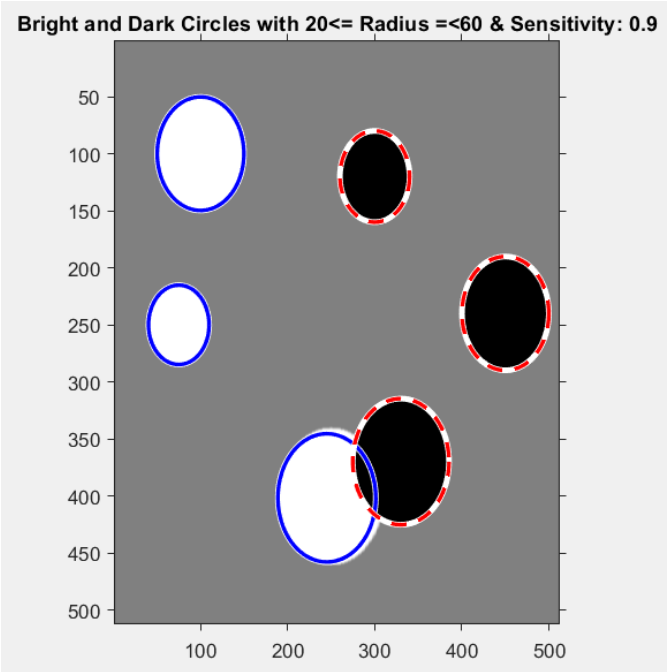




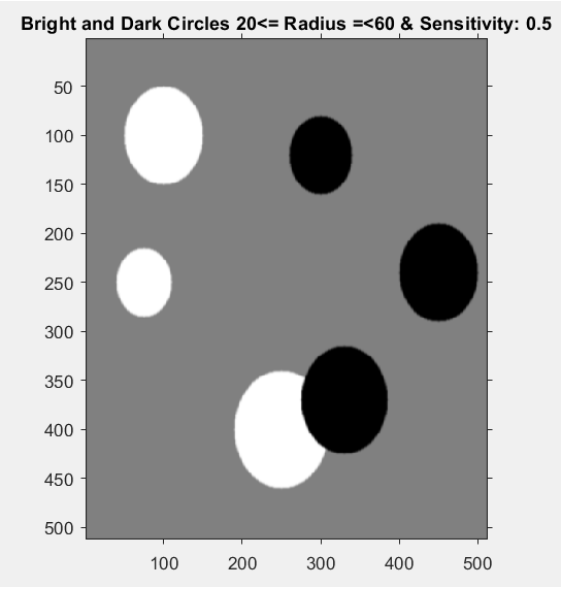


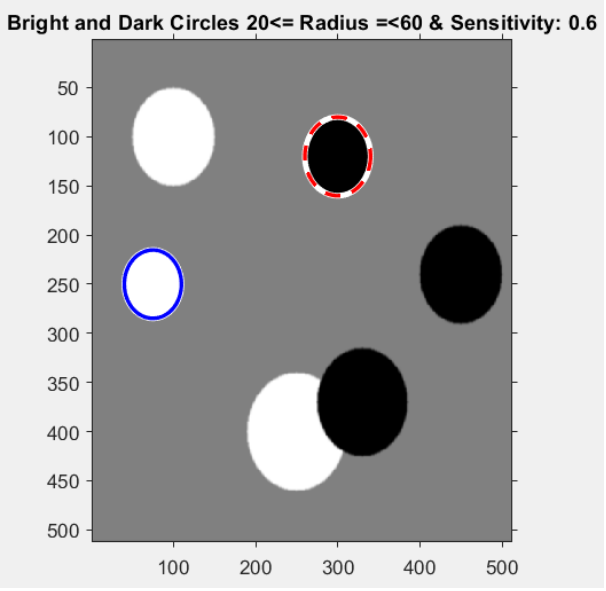
We can say that by increasing the sensitivity factor, we can detect more circles (compare 0.6 and 0.7 and 0.9) in the image. Also, we see that 0.5 is shown a low sensitivity factor that could not detect any circles in the image. On the other hand, sensitivity of a 0.99 resulted in detecting unwanted circler shapes that are supposed as circles by the program. Therefore, we can understand that a sensitivity factor about 0.8-0.9 will give a good performance. Moreover, it gives us a understanding of why built-in Sensitivity factor in MATLAB environment is 0.85.

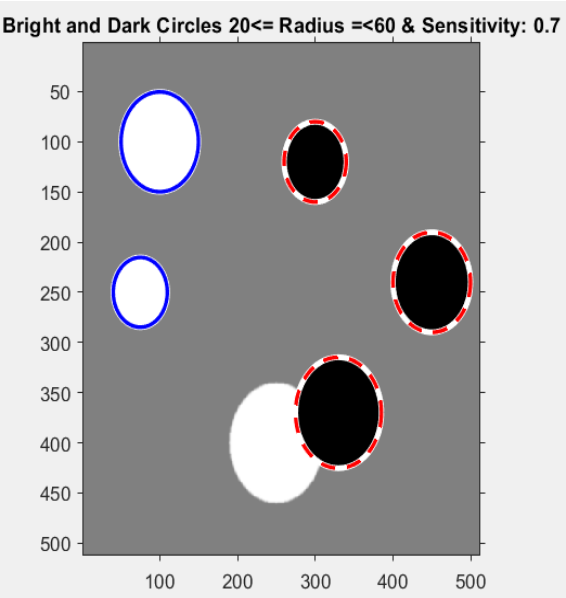
Lastly, we also tried to detect the darker circles in the image (on the first steps we detected only the brighter ones). (Elapsed time: 0.202398 seconds)

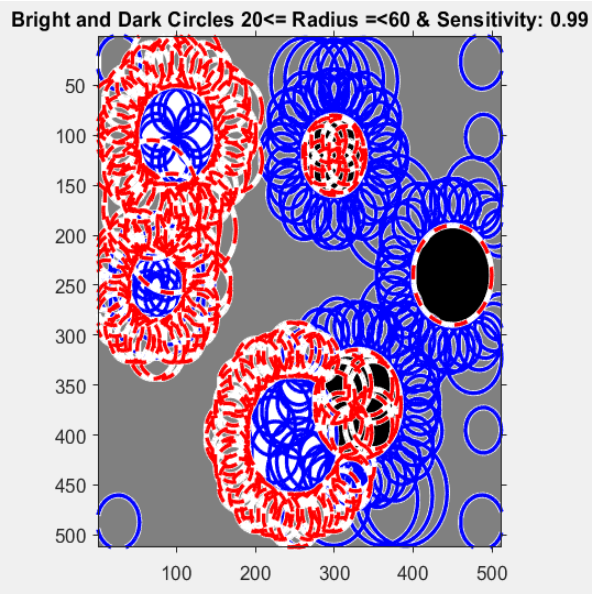


Again we varied the sensitivity factor of the image to see the effects.









Again, we can say that by increasing the sensitivity factor, we can detect more circles (compare 0.6 and 0.7 and 0.9) in the image. Also, we see that 0.5 is shown a low sensitivity factor that could not detect any circles in the image. Again, sensitivity of a 0.99 resulted in detecting unwanted circler shapes that are supposed as circles by the program.

lab4houghcircles.m

```
function [centersBright, radiiBright, centersDark,
radiiDark] = lab4houghlines(img,Rmin,Rmax,Sensitivity)

    [row,col,ch] = size(img);
    if(ch==3)
        img = rgb2gray(img);
    end
    tic;
    [centersBright, radiiBright] = imfindcircles(img,[Rmin
Rmax],'ObjectPolarity','bright','Sensitivity',Sensitivity)
;
    [centersDark, radiiDark] = imfindcircles(img,[Rmin
Rmax],'ObjectPolarity','dark','Sensitivity',Sensitivity);
    toc;
end
```

lab4.m (second part)

```
img = imread('checker.jpg');

 [row,col,ch] = size(img);
 if(ch==3)
    img = rgb2gray(img);
 end

circleimg = imread('circlesBrightDark.png');

subplot(1,3,1);
imshow(circleimg);
title('Circles 20<= Radius =<60');
[centersBright, radiiBright, centersDark, radiiDark] =
lab4houghcircles(circleimg,20,60,0.85);
axis on, axis normal, hold on;
viscircles(centersBright, radiiBright,'Color','b');
axis on, axis normal, hold on;

subplot(1,3,2);
imshow(circleimg);
title('Circles 20<= Radius =<60 & Sensitivity: 0.9');
[centersBright, radiiBright, centersDark, radiiDark] =
lab4houghcircles(circleimg,20,60,0.9);
axis on, axis normal, hold on;
viscircles(centersBright, radiiBright,'Color','b');
axis on, axis normal, hold on;

subplot(1,3,3);
imshow(circleimg);

title('Bright and Dark Circles 20<= Radius =<60 &
Sensitivity: 0.99');
[centersBright, radiiBright, centersDark, radiiDark] =
lab4houghcircles(circleimg,20,60,0.99);
axis on, axis normal, hold on;
viscircles(centersBright, radiiBright,'Color','b');
axis on, axis normal, hold on;
viscircles(centersDark, radiiDark,'LineStyle','--');
axis on, axis normal, hold on;
```