INTRODUCTION

Corner points of calibration objects detected by using two different methods: Harris Corners and Intersection point of two lines in this lab.

During the in-lab assignment, corner points of checkerboard sticked to a cube which is used for calibration of the camera is investigated deeply.

In the post-lab part of this report, my own calibration object will be passed through similar processes to see performance of two these two different methods.

At the end of the report, discussion will be provided about these two corner detection methods and their results.

1. In-Lab Part of the Report

Following steps applied to Calibration Object one by one:

- Read image, convert it to the black-white edge image. (Apply Smoothing before edge detection)
- Find lines in the Edge image with built-in functions of MATLAB (hough, houghpeaks, houghlines)
- Show these lines in the image with specific colors
- Select two intersecting lines, find their equations and intersection point
- Find Harris Corners with blue circles
- Compare their distances between them and with the original corner coordinate

I have chosen to use Canny as the Edge detector as it has good performance results in the previous lab. Then, I applied Hough Transform to go from image space to parameter space where each points represented as sinusoids. Then, I found the peaks in this parameter space. These peaks used to determine lines by using houghlines built-in function.

After finding lines, I have chosen two intersecting lines manually. Following is the general form of the sinusoid:

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

From this equation I can write my two intersecting lines equation as following way:

120 = x*cos(-8) + y*sin(-8) and -108 = x*cos(-69.5) + y*sin(-69.5)

Then, I found the intersection point by solving these equations as following way:

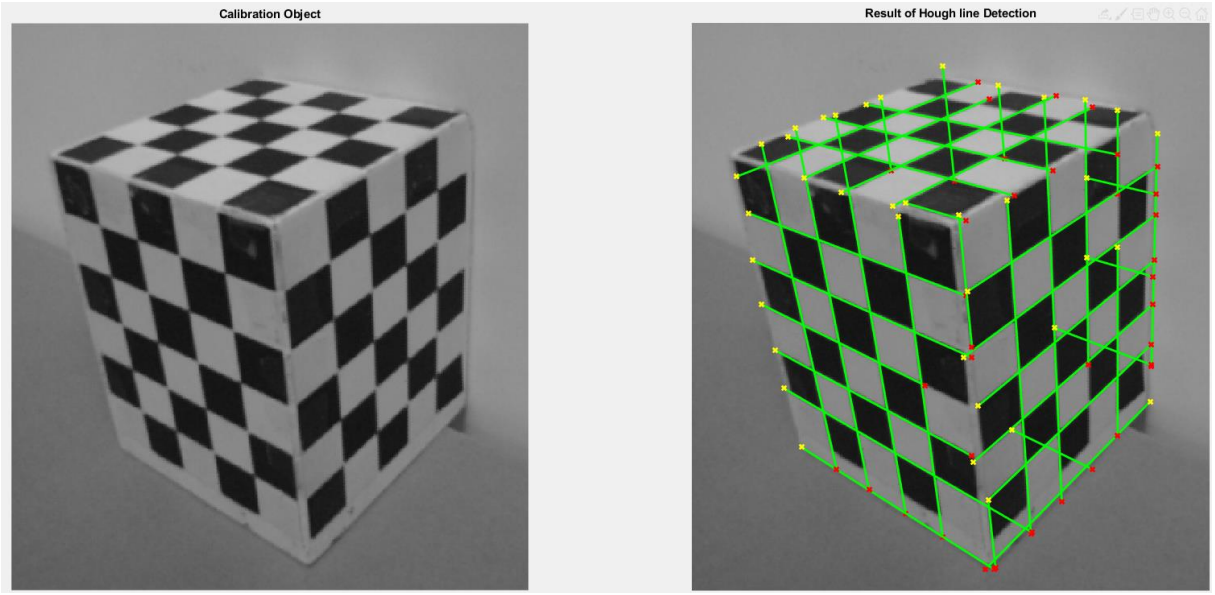
Intersection point [x; y] = $[\cos(\theta_1), \sin(\theta_1); \cos(\theta_2), \sin(\theta_2)]^{-1}$ $[\rho_1, \rho_2]$

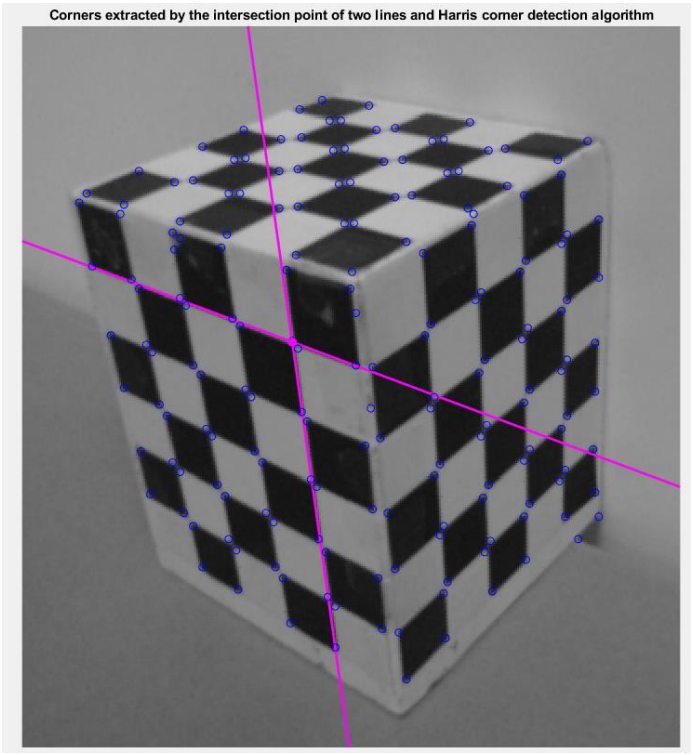
Then, I also found the corners with Harris Detector. At the end, I compared their distances.

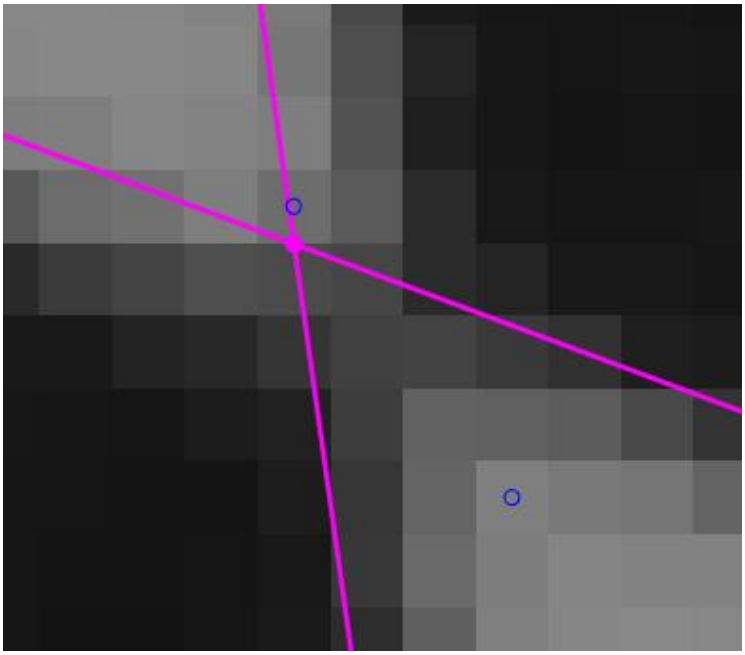
Results can be seen in the following part.

Ege Erdoğan 25331

RESULTS







```
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 0.51632
Distance between Actual Corner Point and intersection point (magenta circle): 1.7874
Distance between Actual Corner Point and Harris Corner (blue circle): 2.2361
```

2. Post-Lab Part of the Report

In the post-lab, same steps with in-lab will be followed. Only difference is this time we will extract 8 corner points. Discussion will be provided at the end of the report.

Extracted lines and their equations: I picked 3 horizontal and 3 vertical lines with their 9 intersecting points. Therefore, I have 6 line equations with 9 intersections.

Line's equation can be seen as follow: $\mathbf{y = \rho - x*\cos(\theta) / \sin(\theta)}$

$Y_1 = -287 - x*\cos(-81) / \sin(-81)$

$Y_2 = -309 - x*\cos(-77.5) / \sin(-77.5)$

$Y_3 = -266 - x*\cos(-85) / \sin(-85)$

$Y_4 = 271 - x*\cos(0.5) / \sin(0.5)$

$Y_5 = 231 - x*\cos(-0.5) / \sin(-0.5)$

$Y_6 = 314 - x*\cos(1) / \sin(1)$

Intersection points extracted via following way:

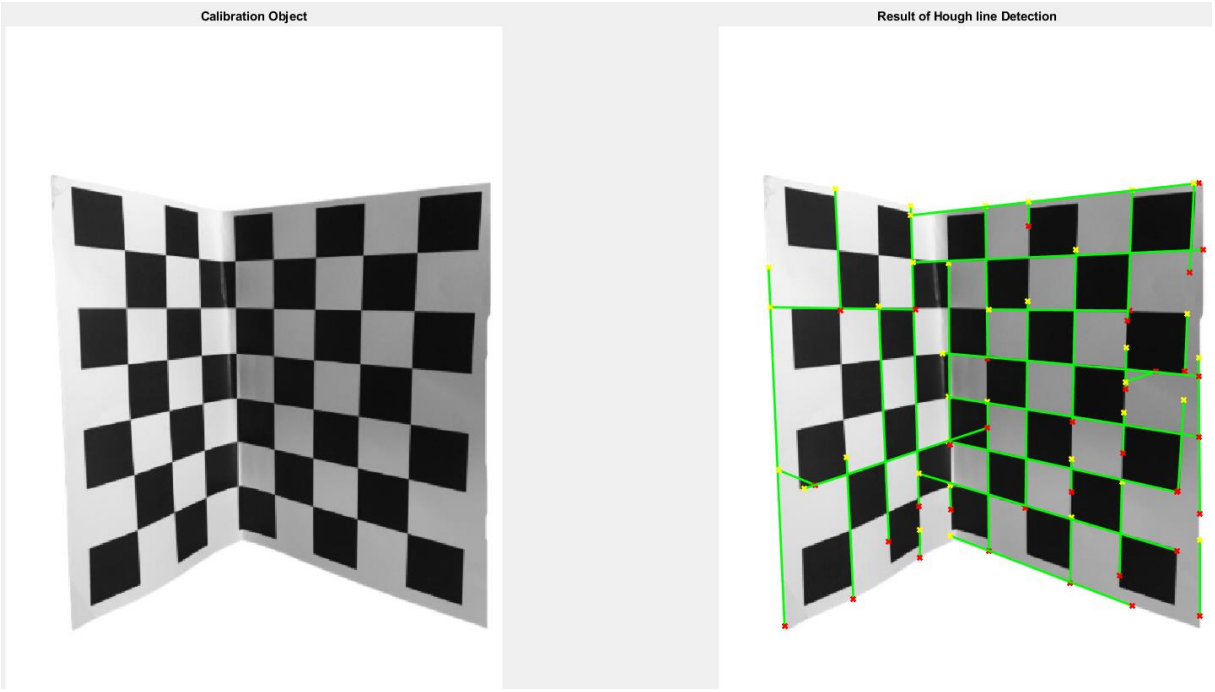
Intersection point [x; y] = $[\cos(\theta_1), \sin(\theta_1); \cos(\theta_2), \sin(\theta_2)]^{-1} [\rho_1, \rho_2]$

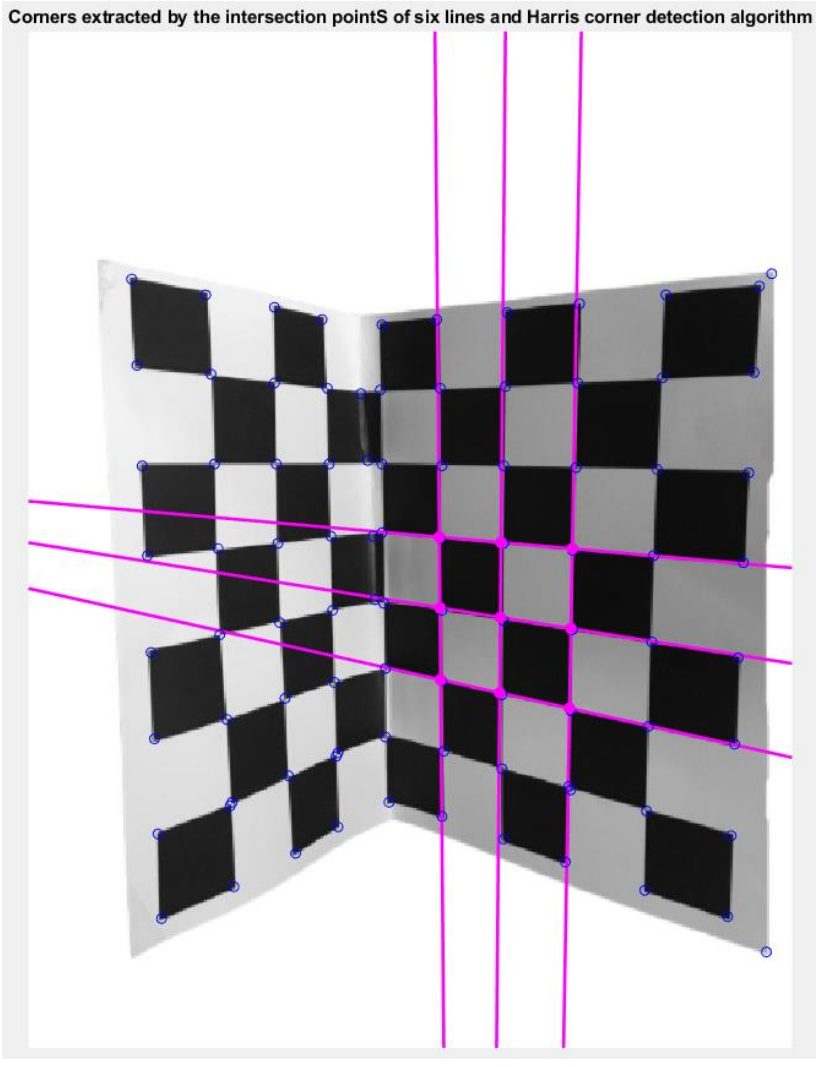
Intersection points list:

9x2 double

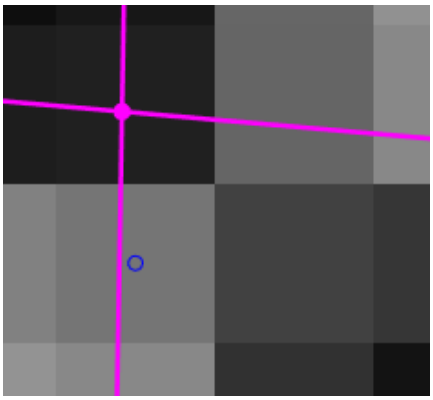
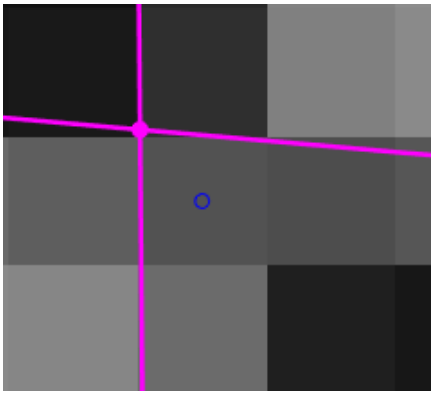
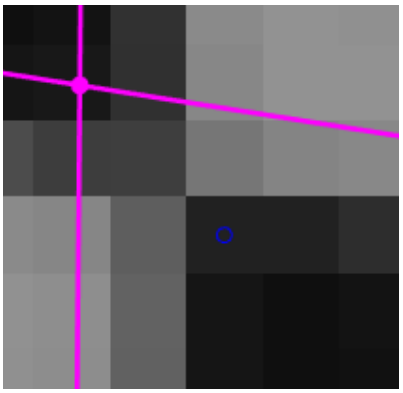
	1	2
1	268.1039	333.0410
2	233.8679	327.6185
3	308.1239	339.3795
4	267.7303	375.8567
5	234.2240	368.4286
6	307.3340	384.6367
7	268.4751	290.5046
8	233.5173	287.4462
9	308.9153	294.0427

RESULTS


Calibration Object
Result of Hough line Detection


Corners extracted by the intersection pointS of six lines and Harris corner detection algorithm

Zoomed Versions for some corners:

```
Corner Point: 1
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 2.7263
Distance between Actual Corner Point and intersection point (magenta circle): 2.0194
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 2
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 1.7861
Distance between Actual Corner Point and intersection point (magenta circle): 0.40371
Distance between Actual Corner Point and Harris Corner (blue circle): 1.4142
Corner Point: 3
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 1.0735
Distance between Actual Corner Point and intersection point (magenta circle): 1.1819
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 4
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 1.7086
Distance between Actual Corner Point and intersection point (magenta circle): 1.0031
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 5
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 0.96367
Distance between Actual Corner Point and intersection point (magenta circle): 0.96367
Distance between Actual Corner Point and Harris Corner (blue circle): 0
Corner Point: 6
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 1.5173
Distance between Actual Corner Point and intersection point (magenta circle): 0.87915
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 7
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 1.6033
Distance between Actual Corner Point and intersection point (magenta circle): 1.4287
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 8
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 0.73464
Distance between Actual Corner Point and intersection point (magenta circle): 1.4409
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
Corner Point: 9
Distance between Harris Corner (blue circle) and intersection point (magenta circle): 0.96108
Distance between Actual Corner Point and intersection point (magenta circle): 0.74232
Distance between Actual Corner Point and Harris Corner (blue circle): 0.70711
```

DISCUSSION

We used two different methods for detecting corners in our calibration objects, Harris and intersection points of two lines with the help of Hough Transform.

We can observe that we can find corners with the help of these two methods. However, when we have non accurate lines such as skewed lines at the end of Hough Transform and line detection, we are ending up with miss placed corners in the image. In other words, detected corner points are far away from the original corner point because of wrong alignment of Hough lines.

We can observe that Harris method results with good performance on both number of detected corner point and also accuracy of these points. Compared to Harris, we can end up with additional corner points if we find each intersection points of detected lines. However, I think that accuracy of Harris method is better than the intersection method for my case.

Normally, line intersection with Hough Transform should have higher accuracy as it has subpixel accuracy (having real number coordinates). However, I think my photography skill was not good enough to take good photo for detecting lines with Hough Transform 😃.

Consequently, I prefer to use Harris method as its higher accuracy and easy usage (one line code with the help of MATLAB) in my calibration.

According to Corner Detection Techniques: An Introductory Survey, there are many of different corner detection methods. For instance, Susan Corner Detector is using circular mask around each pixel where the center pixel is compared to other pixels that is in the mask. Then, comparing the reduction of area of mask, edges and corners can be detected fast and with good repeatability rate. However, it is sensitive to noise.

Another method can be the The Features from Accelerated Segment Test (FAST) which detects only corner points with the help of machine learning. According to Comparative Study of Corner and Feature Extractors for Real-Time Object Recognition in Image Processing, this method is faster, have better repeatability error.

REFERENCES

Mohapatra, Arpita & Sarangi, Sunita & Patnaik, Srikanta & Sabut, Sukanta. (2014). Comparative Study of Corner and Feature Extractors for Real-Time Object Recognition in Image Processing. Journal of information and communication convergence engineering. 12. 263-270. 10.6109/jicce.2014.12.4.263.

Patel, T., & Panchal, S.R. (2014). Corner Detection Techniques: An Introductory Survey.

CODES

```
object = imread("mychecker.png");
% Ensuring grey scale image
[row,col,ch] = size(object);
if(ch == 3)
    img = rgb2gray(object);
end
img = imgaussfilt(img);
BW = edge(img,'canny');
[H,T,R] = hough(BW,'Theta',-90:0.5:89);

peaks = 20;
P  = houghpeaks(H,peaks);

% Find lines and plot them
fillgap = 20;
minlength = 20;
lines = houghlines(BW,T,R,P,'FillGap',fillgap,'MinLength',minlength);
figure()
subplot(1,2,1)
imshow(img);
title("Calibration Object")
subplot(1,2,2)
imshow(img);
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');
end
title("Result of Hough line Detection")

thetas = [-81, -77.5, -85, 0.5, -0.5, 1]; %first 3 horizontal, next 3 vertical
rhos = [-287, -309, -266, 271, 231, 314];
```

```
figure()
imshow(img)
hold on
x = -500:500;
for i = 1:6
    y = (rhos(1, i) - x*cosd(thetas(1, i))) / sind(thetas(1,i));
    plot(x, y,'LineWidth',2,'Color', 'magenta')
end
C = corner(img);
plot(C(:,1),C(:,2),'bo');
intersection_list = [];
for i = 1:3
    for k = 4:6
    B = [rhos(1,i); rhos(1,k)];
    A = [cosd(thetas(1,i)), sind(thetas(1,i)); cosd(thetas(1,k)), sind(thetas(1,k))];
    inv_a = inv(A);
    intersection = inv_a* B;
    intersection_list = [intersection_list; [intersection(1,1), intersection(2,1)]];
    plot(intersection(1,1), intersection(2,1), 'r.', 'Color', 'magenta', 'MarkerSize', 25);
    end
end
title("Corners extracted by the intersection pointS of six lines and Harris corner detection algorithm")
harris_corners = [64, 62, 63, 49, 65, 46, 70, 74, 55];
actual_point_x = [269.5, 234, 308.5, 268.5, 235, 307.5, 269.5, 234.5, 309.5];
actual_point_y = [334.5, 328, 340.5, 376.5, 369, 385.5, 291.5, 288.5, 294.5];
for i = 1:9
    disp("Corner Point: "+i)
    x1 = intersection_list(i,1);
    y1 = intersection_list(i,2);
    x2 = C(harris_corners(1,i),1);
    y2 = C(harris_corners(1,i),2);
    actual_x = actual_point_x(1,i);
    actual_y = actual_point_y(1,i);
    my_distance = sqrt((x1-x2)^2 + (y1-y2)^2);
    my_distance2 = sqrt((x1-actual_x)^2 + (y1-actual_y)^2);
    my_distance3 = sqrt((x2-actual_x)^2 + (y2-actual_y)^2);
    disp("Distance between Harris Corner (blue circle) and intersection point (magenta circle): "+my_distance)
    disp("Distance between Actual Corner Point and intersection point (magenta circle): "+my_distance2)
    disp("Distance between Actual Corner Point and Harris Corner (blue circle): "+my_distance3)
end
```

```
% % For INLAB RESULTS just uncomment below part
% object = imread("calibrationObject.png");
% % Ensuring grey scale image
% [row,col,ch] = size(object);
% if(ch == 3)
%     img = rgb2gray(object);
% end
% BW = edge(img,'canny');
% [H,T,R] = hough(BW,'Theta',-90:0.5:89);
% peaks = 30;
% P  = houghpeaks(H,peaks);
% % Find lines and plot them
% fillgap = 20;
% minlength = 40;
% lines = houghlines(BW,T,R,P,'FillGap',fillgap,'MinLength',minlength);
% figure()
% subplot(1,2,1)
% imshow(img);
% title("Calibration Object")
% subplot(1,2,2)
% imshow(img);
% 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');
% end
% title("Result of Hough line Detection")
```

```
% first_beginning = [141, 132];
% first_ending = [171, 350];
% first_theta = -8;
% first_rho = 120;
% second_beginning = [39, 130];
% second_ending = [187, 186];
% second_theta = -69.5;
% second_rho = -108;
% figure()
% imshow(img)
% hold on
% x = -500:500;
% y1 = (first_rho - x*cosd(first_theta)) / sind(first_theta);
% y2 = (second_rho - x*cosd(second_theta)) / sind(second_theta);
% plot(x, y1 ,'LineWidth',2,'Color', 'magenta')
% plot(x,y2 ,'LineWidth',2,'Color', 'magenta')
% C = corner(img);
% plot(C(:,1),C(:,2),'bo');
%
% B = [first_rho; second_rho];
% A = [cosd(first_theta), sind(first_theta); cosd(second_theta), sind(second_theta)];
% inv_a = inv(A);
% intersection = inv_a* B;
% plot(intersection(1,1), intersection(2,1), 'r.', 'Color', 'magenta', 'MarkerSize', 25);
% x1 = intersection(1,1);
% y1 = intersection(2,1);
% x2 = C(44,1);
% y2 = C(44,2);
% actual_point_x = 146;
% actual_point_y = 171;
% my_distance = sqrt((x1-x2)^2 + (y1-y2)^2);
% my_distance2 = sqrt((x1-actual_point_x)^2 + (y1-actual_point_y)^2);
% my_distance3 = sqrt((x2-actual_point_x)^2 + (y2-actual_point_y)^2);
% disp("Distance between Harris Corner (blue circle) and intersection point (magenta circle): "+my_distance)
% disp("Distance between Actual Corner Point and intersection point (magenta circle): "+my_distance2)
% disp("Distance between Actual Corner Point and Harris Corner (blue circle): "+my_distance3)
% title("Corners extracted by the intersection point of two lines and Harris corner detection algorithm")
```