Student: Melis Kilic

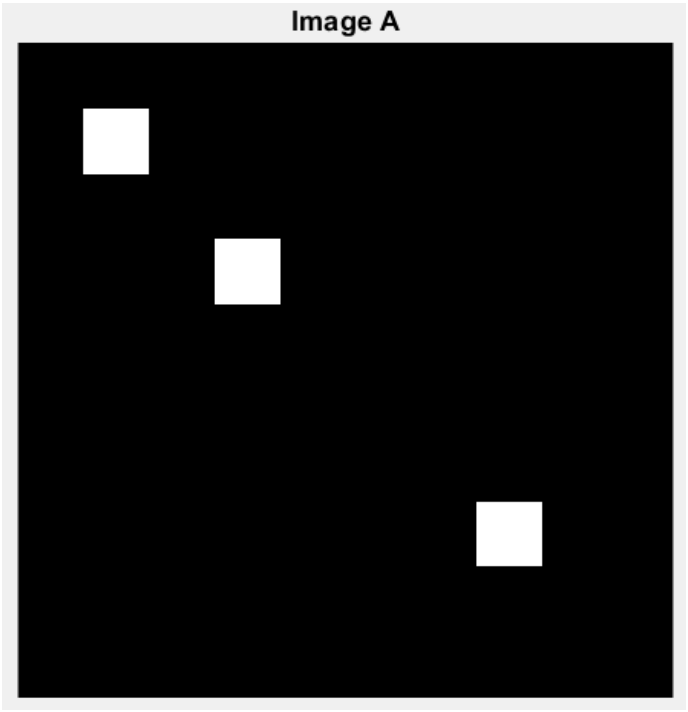
Hough transform

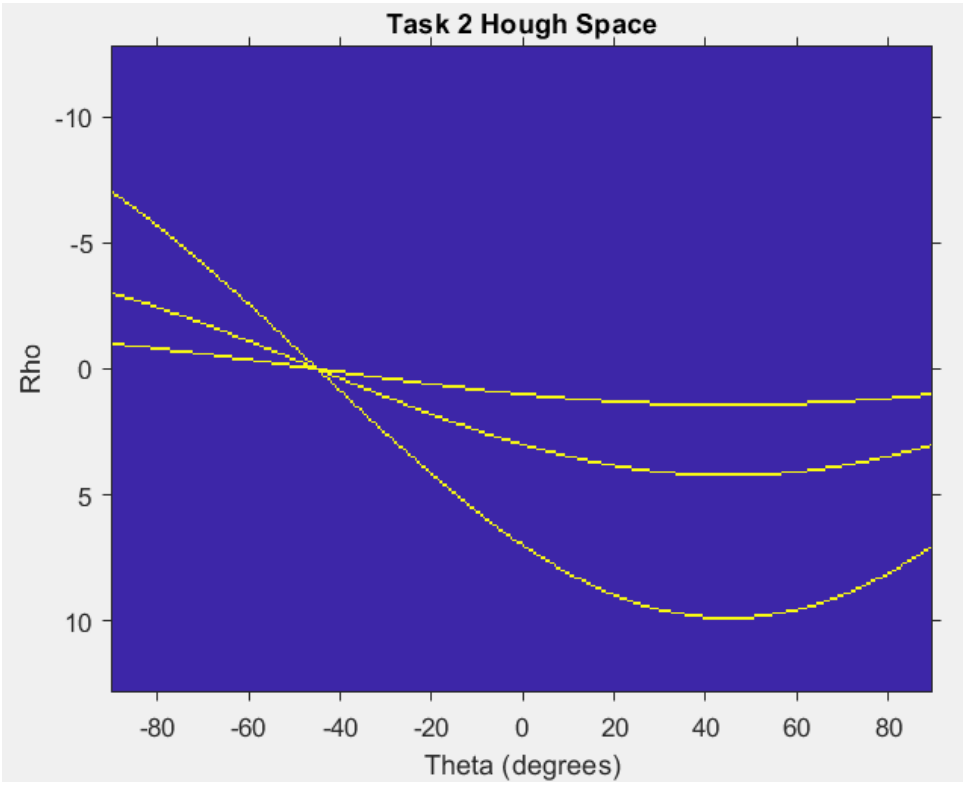
HT-1. Hough transform of the set of colinear points.

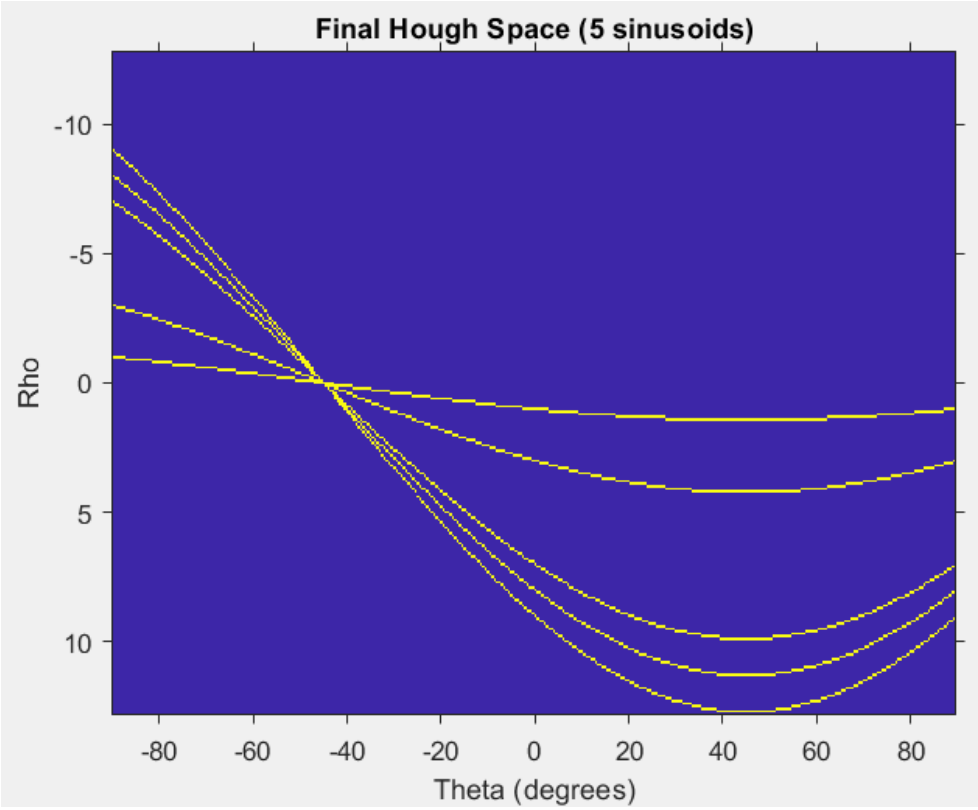
Task 1a Hough space visualization

```
%% Task 1a Hough space visualization

close all;
clearvars;
clc;

% Task 1:
A = zeros(10, 10);
A(2, 2) = 1;
A(4, 4) = 1;
A(8, 8) = 1;

% Display
figure;
imagesc(A);
title('Image A');
colormap(gray);
axis equal;
axis off;

% Task 2:
theta = -90:0.5:89.5;
[H, theta, rho] = hough(A, 'Theta', theta, 'RhoResolution', 0.1);

% Display
figure;
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit', 'Colormap', parula);
title('Task 2 Hough Space');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;

%  Task 3:
A(9, 9) = 1;
A(10, 10) = 1;

[H, theta, rho] = hough(A, 'Theta', theta, 'RhoResolution', 0.1);

% Display
figure;
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit', 'Colormap', parula);
title('Final Hough Space (5 sinusoids)');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
```

Result of the code:







Based on theory from the lecture answer the questions:

- ***What is represented by the axes of the Hough space image?***

Axes of the Hough space image represent the parameters of the Hough transform: theta (angle) and rho (distance from the origin).

- ***Which sinusoid corresponds to which pixel of the image and why?***

Each sinusoid corresponds to a line in the image space passing through a point. The intersection point in the Hough space represents the parameters (theta, rho) of that line.

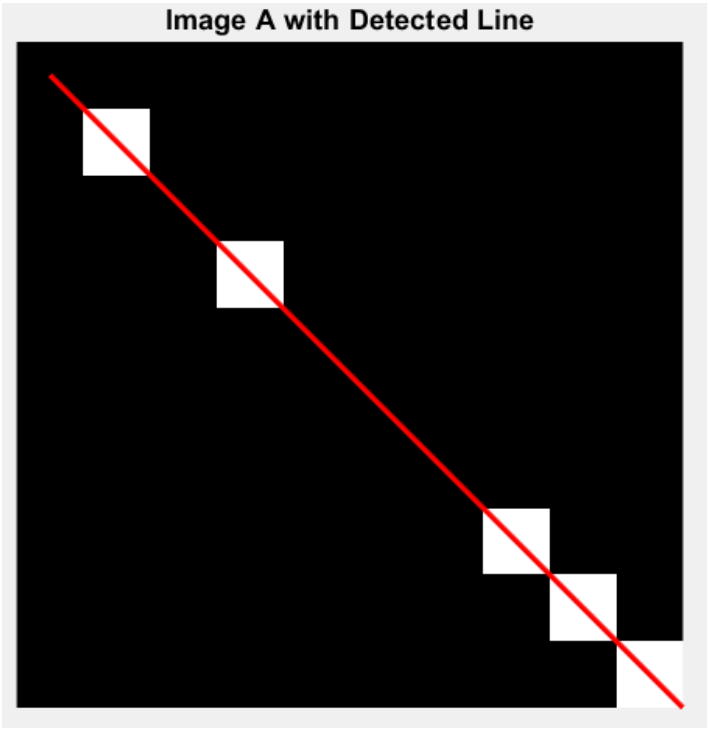
- ***Why do all three sinusoids intersect at one point?***

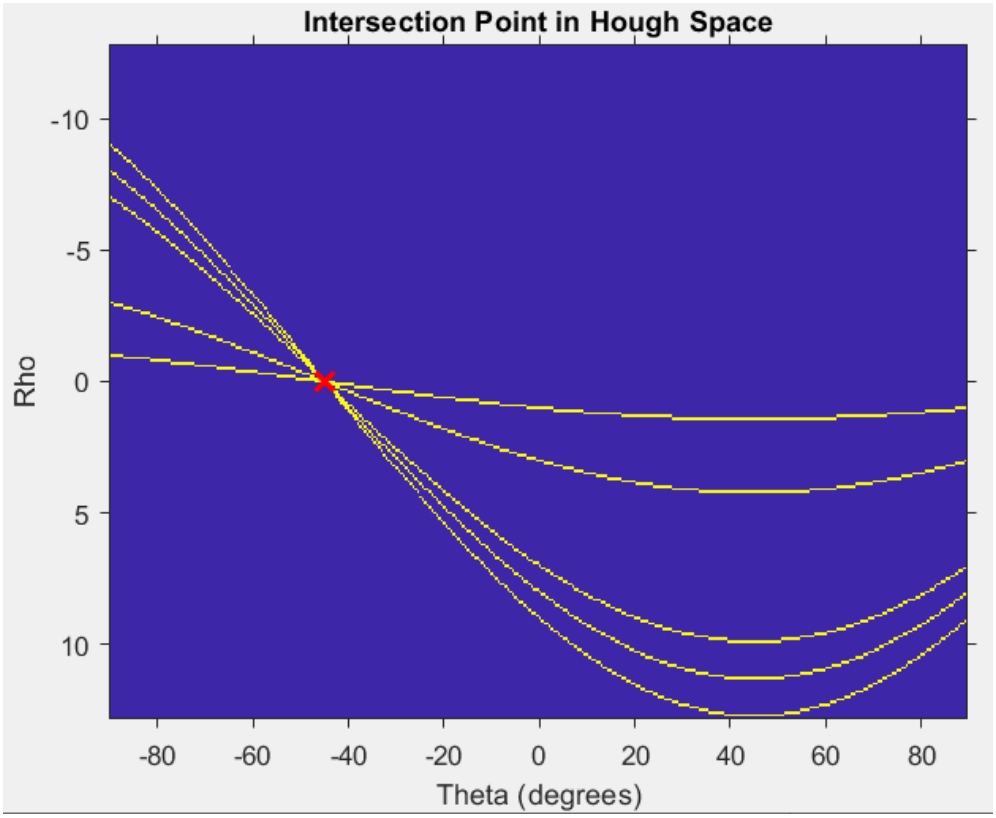
Three colinear pixels in the image result in three sinusoids intersecting at one point because they lie on the same line in the image space.

Task 1b Parameters of the straight line

```
%% Task 1b Parameters of the straight line

% Task 4:
[maxValue, maxIndex] = max(H(:));
[row, col] = ind2sub(size(H), maxIndex);

% Display
figure;
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit', 'Colormap', parula);
title('Intersection Point in Hough Space');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
hold on;
plot(theta(col), rho(row), 'rx', 'MarkerSize', 10, 'LineWidth', 2);

% Task 5:
x = 0:0.1:10;
thetaRadians = deg2rad(theta(col));
y = (rho(row) - x .* cos(thetaRadians)) / sin(thetaRadians);

% Task 6:

% Display
Figure;
imagesc(A);
title('Image A with Detected Line');
colormap(gray);
axis equal;
axis off;
hold on;
plot(x + 1, y + 1, 'r', 'LineWidth', 2);
hold off;
```

Result of the code:





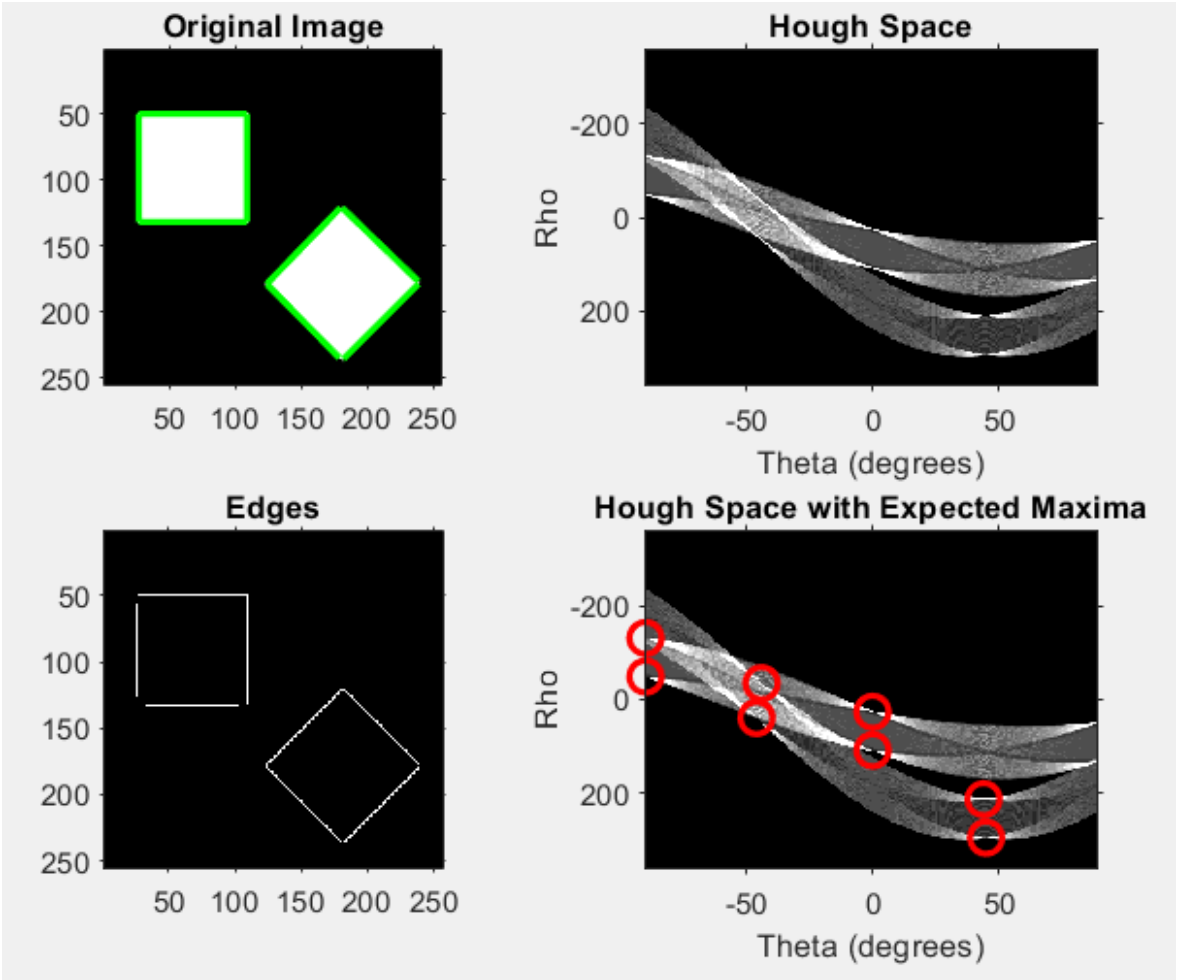
HT-2. Hough transform of the set of non-colinear points.

HT-3. Hough transform of an artificial image “squares”

```
close all;
clearvars;
clc;

originalImage = imread('squares.png');
edgesImage = edge(originalImage, 'Canny');
[H, theta, rho] = hough(edgesImage);

% Display results in subplots
figure('Name', 'Hough Transform Results');
colormap(gray);

subplot(2, 2, 1);
imshow(originalImage);
title('Original Image');
axis on;

subplot(2, 2, 2);
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit');
title('Hough Space');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
hold on;

subplot(2, 2, 3);
imshow(edgesImage);
title('Edges');
axis on;

subplot(2, 2, 4);
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit');
title('Hough Space with Expected Maxima');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
hold on;

% Use houghpeaks to look for 8 maxima
peaks = houghpeaks(H, 8, 'NHoodSize', [19, 19]);

% Plot maxima in Hough space
plot(theta(peaks(:, 2)), rho(peaks(:, 1)), 'o', 'MarkerSize', 10,
'MarkerEdgeColor', 'r', 'LineWidth', 2);

% Find and display lines in the image using houghlines
lines = houghlines(edgesImage, theta, rho, peaks);
```

```
% Display the lines on the original image
subplot(2, 2, 1);
hold on;
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:, 1), xy(:, 2), 'LineWidth', 2, 'Color', 'g');
end
hold off;

% Display the lines in the Hough space with maxima marked
subplot(2, 2, 2);
for k = 1:length(lines)
    plot(lines(k).theta, lines(k).rho, 'g', 'LineWidth', 2);
end
hold off;
```

Result of the code:

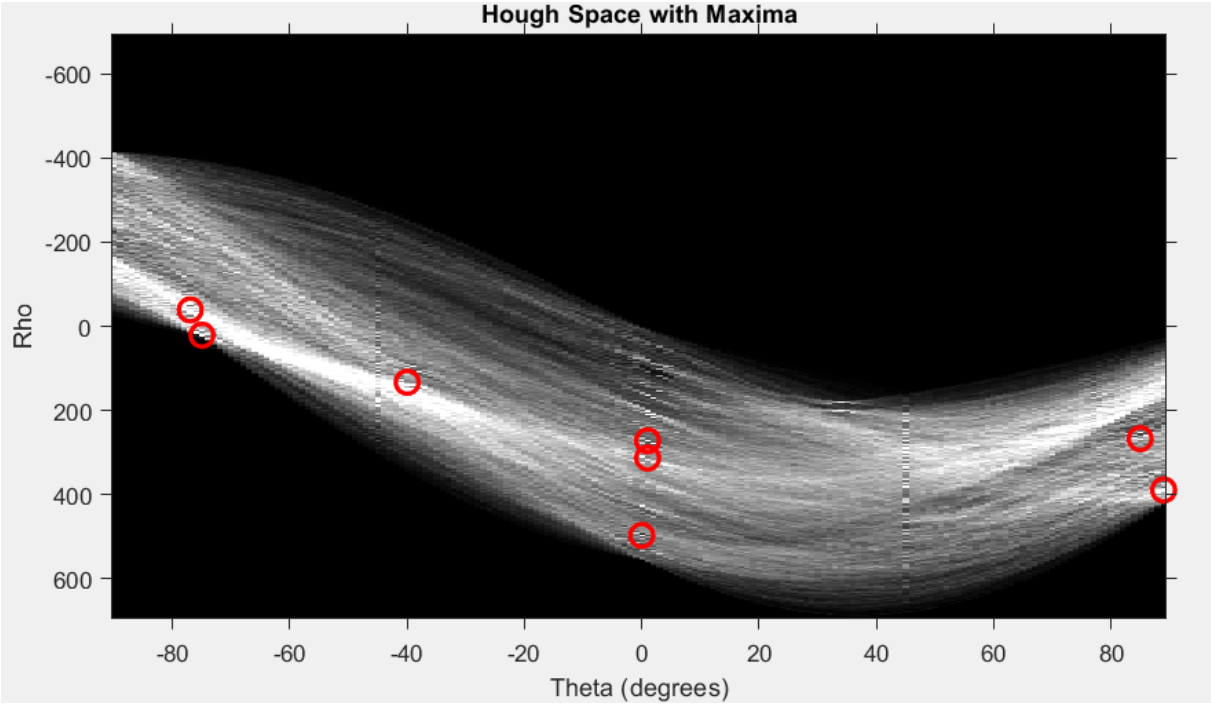


HT-4. Hough transform of a real-life image “house”

```
close all;
clearvars;
clc;

originalImage = imread('house.png');

% Convert to grayscale if the image is in RGB format
if size(originalImage, 3) == 3
    grayImage = rgb2gray(originalImage);
else
    grayImage = originalImage;
end

edgesImage = edge(grayImage, 'Canny', [], 1.5);
[H, theta, rho] = hough(edgesImage);

% Display
figure('Name', 'Hough Transform Results for house.png');
colormap(gray);

% Original image
subplot(1, 3, 1);
imshow(originalImage);
title('Original Image');
axis on;

% Edges with houghlines superimposed
subplot(1, 3, 2);
imshow(originalImage);
title('Edges with Houghlines');
axis on;
hold on;

% Find and display lines in the image using houghlines
peaks = houghpeaks(H, 8, 'NHoodSize', [19, 19]);
lines = houghlines(edgesImage, theta, rho, peaks);
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:, 1), xy(:, 2), 'LineWidth', 2, 'Color', 'g');
end
hold off;

% Edges
subplot(1, 3, 3);
imshow(edgesImage);
title('Edges');
axis on;

% Hough space with maxima marked
figure('Name', 'Hough Space with Maxima');
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit');
title('Hough Space with Maxima');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
```

```
hold on;
plot(theta(peaks(:, 2)), rho(peaks(:, 1)), 'o', 'MarkerSize', 10,
'MarkerEdgeColor', 'r', 'LineWidth', 2);
hold off;
```

Result of the code:



HT-5. Hough transform of real-life image with morphological operations

```
close all;
clearvars;
clc;

realImage = imread('real_squares.png');

% Convert to grayscale if the image is in RGB format
if size(realImage, 3) == 3
    grayImage = rgb2gray(realImage);
else
    grayImage = realImage;
end

% Apply filtering, morphological processing, binarization, and edge detection
threshold = 0.5;
binaryImage = imbinarize(grayImage, threshold);
se = strel('square', 5);
closedImage = imclose(binaryImage, se);
edgesImage = edge(closedImage, 'Canny');

[H, theta, rho] = hough(edgesImage);
peaks = houghpeaks(H, 8, 'NHoodSize', [19, 19]);

% Display
figure('Name', 'Hough Transform Results for real_squares.png', 'Position', [100,
100, 1200, 400]);

% Original image
subplot(1, 3, 1);
imshow(realImage);
title('Original Image');
axis on;

% Processed image
subplot(1, 3, 2);
imshow(closedImage);
title('Processed Image');
axis on;

% Edges with houghlines superimposed
subplot(1, 3, 3);
imshow(realImage);
title('Edges with Houghlines');
axis on;
hold on;

% Find and display lines in the image using houghlines
lines = houghlines(edgesImage, theta, rho, peaks);
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:, 1), xy(:, 2), 'LineWidth', 2, 'Color', 'g');
end
hold off;
```

```
% Hough space with maxima marked (in a separate figure)
figure('Name', 'Hough Space with Maxima', 'Position', [300, 100, 600, 400]);
imshow(imadjust(mat2gray(H)), 'XData', theta, 'YData', rho,
'InitialMagnification', 'fit');
title('Hough Space with Maxima');
xlabel('Theta (degrees)');
ylabel('Rho');
axis on;
axis normal;
hold on;
plot(theta(peaks(:, 2)), rho(peaks(:, 1)), 'o', 'MarkerSize', 10,
'MarkerEdgeColor', 'r', 'LineWidth', 2);
hold off;
```

Result of the code: