ASSIGNMENT - 3
CPS 667 : ADVANCED COMPUTER VISION

DARSHAN JIGALA CHANNA REDDY
101792933

```
% Load the image
originalImage = imread('cameraman.tif');
% Convert the image to grayscale if it's not already
if size(originalImage, 3) == 3
   originalImage = rgb2gray(originalImage);
end
% Detect edges using the Canny method
edges = edge(originalImage, 'canny');
% Perform Hough transform
[houghSpace, angleValues, distanceValues] = hough(edges);
% Find the peaks in the Hough transform matrix
peakCount = 5; % Maximum number of peaks to detect
peakThreshold = ceil(0.3 * max(houghSpace(:))); % Threshold for peaks
detectedPeaks = houghpeaks(houghSpace, peakCount, 'threshold', peakThreshold);
% Extract lines from the image
lineGap = 8; % Maximum gap to connect line segments
minimumLineLength = 10; % Minimum length of a line to be detected
lines = houghlines(edges, angleValues, distanceValues, detectedPeaks, ...
                   'FillGap', lineGap, 'MinLength', minimumLineLength);
% Display the original image in a separate figure
figure;
imshow(originalImage);
title('Original Image');
% Display the original image with detected lines in another figure
figure;
imshow(originalImage);
hold on;
% Calculate the lengths of detected lines
lineLengths = arrayfun(@(line) norm(line.point1 - line.point2), lines);
% Sort the lengths of lines in descending order
[~, sortedIndices] = sort(lineLengths, 'descend');
% Draw the 4 longest lines
linesToDisplay = 4; % Number of lines to display
for idx = 1:min(linesToDisplay, length(lines))
   % Get the current line
   currentLine = lines(sortedIndices(idx));

   % Extract the coordinates of the line endpoints
   lineCoordinates = [currentLine.point1; currentLine.point2];

   % Plot the line
   plot(lineCoordinates(:, 1), lineCoordinates(:, 2), 'g', 'LineWidth', 2);

   % Plot the start and end points of the line
   plot(lineCoordinates(1, 1), lineCoordinates(1, 2), 'yx', 'LineWidth', 2); % Yellow cross
   plot(lineCoordinates(2, 1), lineCoordinates(2, 2), 'rx', 'LineWidth', 2); % Red cross
end
% Add title and hold off
title('Longest 4 Lines Detected');
hold off;
```

Output:


Figure 1
Figure 2
Original Image

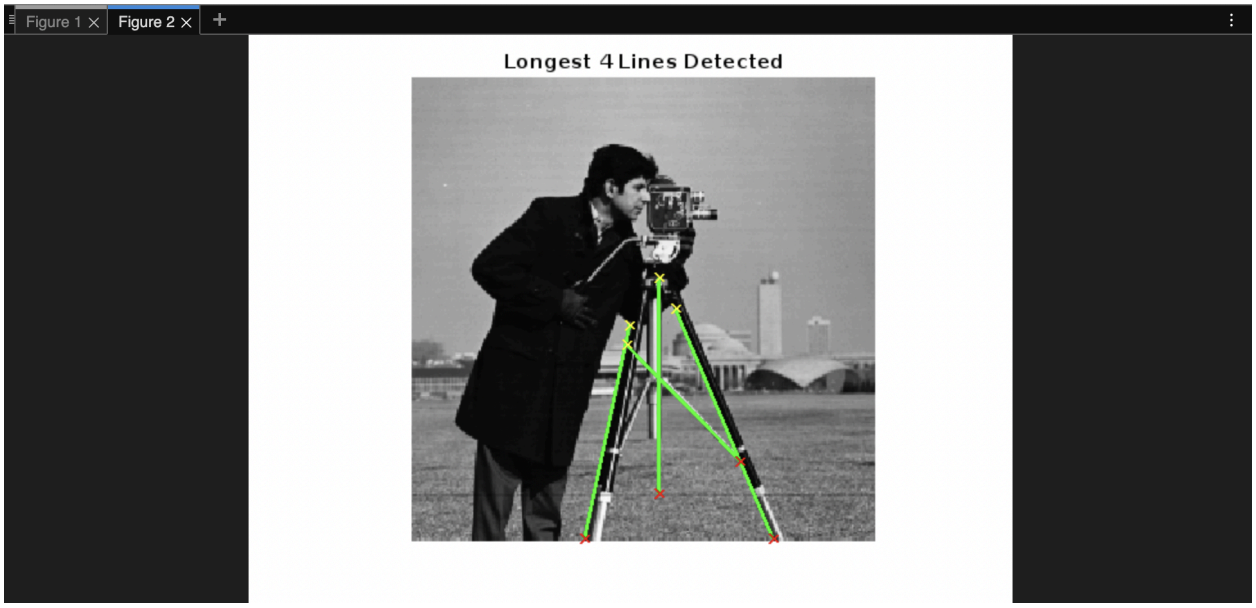
Figure 1
Figure 2
Longest 4 Lines Detected

```
% Load the image
originalImage = imread('rice.png');
% Ensure the image is grayscale
if size(originalImage, 3) == 3
   originalImage = rgb2gray(originalImage);
end
% Detect edges using the Canny method
edges = edge(originalImage, 'canny');
% Perform Hough transform
[houghSpace, angleValues, distanceValues] = hough(edges);
% Find the peaks in the Hough transform matrix
numPeaks = 5; % Number of peaks to detect
threshold = ceil(0.3 * max(houghSpace(:))); % Threshold for peaks
peaks = houghpeaks(houghSpace, numPeaks, 'threshold', threshold);
% Extract lines using the Hough transform
gapFill = 8; % Maximum gap between line segments
minLineLength = 10; % Minimum length of detected lines
lines = houghlines(edges, angleValues, distanceValues, peaks, ...
                   'FillGap', gapFill, 'MinLength', minLineLength);
% Display the original image
figure;
imshow(originalImage);
title('Original Image');
% Display the original image with detected lines
figure;
imshow(originalImage);
hold on;
% Calculate lengths of the detected lines
lineLengths = arrayfun(@(line) norm(line.point1 - line.point2), lines);
% Sort the lengths of the lines in descending order
[~, sortedIndices] = sort(lineLengths, 'descend');
% Draw the 4 longest lines
numLinesToDisplay = 4; % Number of lines to display
for idx = 1:min(numLinesToDisplay, length(lines))
   % Get the current line
   currentLine = lines(sortedIndices(idx));

   % Extract the coordinates of the line endpoints
   lineCoordinates = [currentLine.point1; currentLine.point2];

   % Plot the line
   plot(lineCoordinates(:, 1), lineCoordinates(:, 2), 'g', 'LineWidth', 2);

   % Plot the start and end points of the line
   plot(lineCoordinates(1, 1), lineCoordinates(1, 2), 'yx', 'LineWidth', 2); % Yellow cross
   plot(lineCoordinates(2, 1), lineCoordinates(2, 2), 'rx', 'LineWidth', 2); % Red cross
end
% Add title and hold off
title('Longest 4 Lines Detected');
hold off;
```

Output:

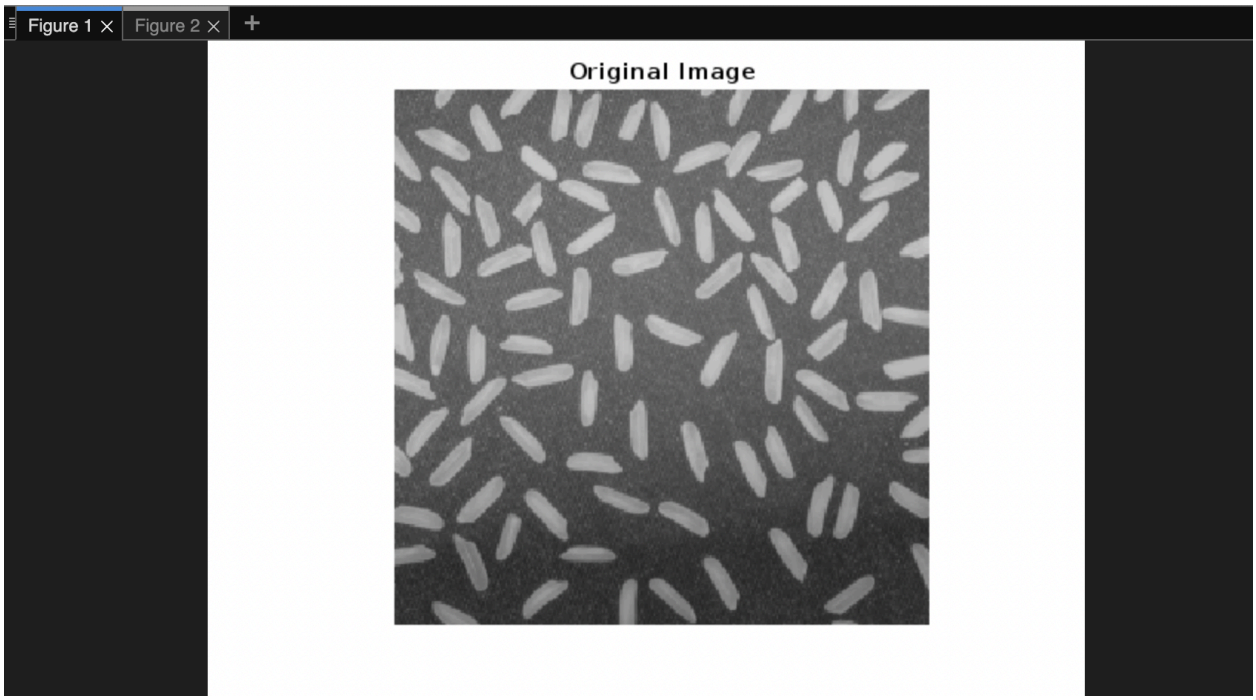
Figure 1
Figure 2
Original Image

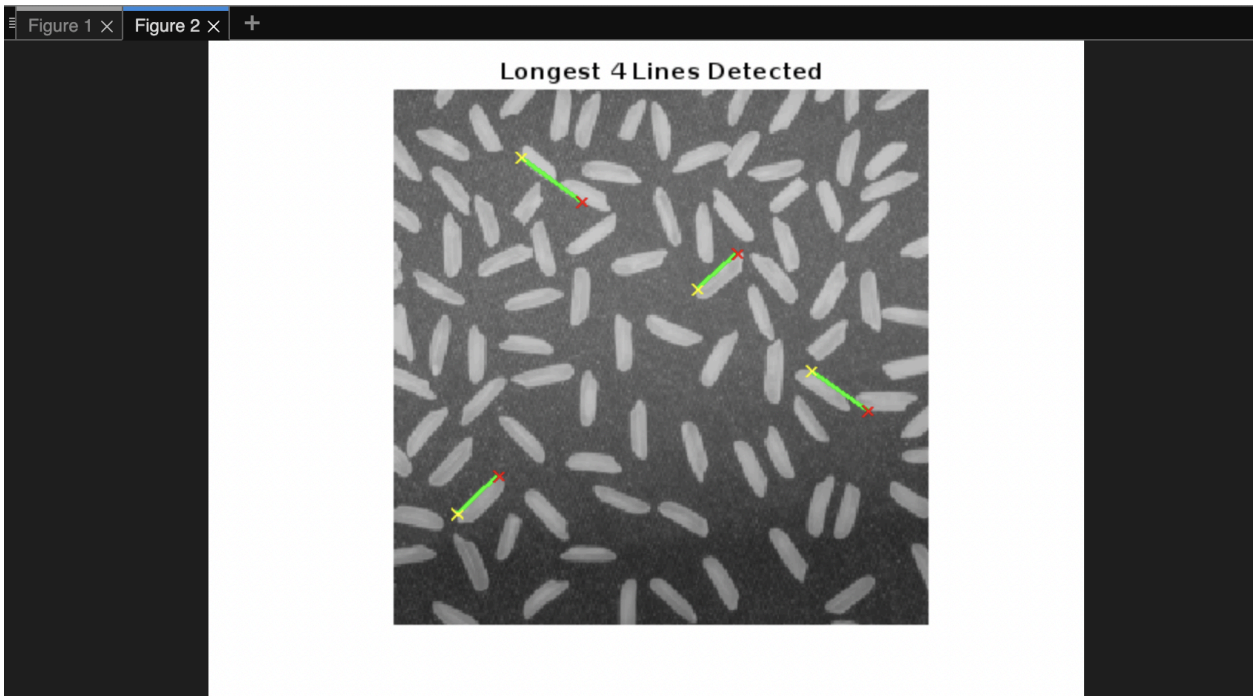
Figure 1
Figure 2
Longest 4 Lines Detected