1 Pre-Recorded Tasks

1.1 Hough Transfrom

As the result of an edge detection algorithm we receive the following five points in the space of a Cartesian coordinate system:

$$x(k) = \begin{bmatrix} 0.5 & 1.0 & 1.5 & 2.0 & 2.5 \end{bmatrix}$$
$$y(k) = \begin{bmatrix} 1.5 & 1.0 & 0.5 & 1.0 & 1.5 \end{bmatrix}$$

1. Plot the detected edges in a diagram.
2. We want to use the Hough Transformation to identify straight lines in the image. For our calculations we use the Hesse normal form following page (6-35) of the script. In order to decrease the amount of calculations we can assume that a line goes through at least two edge points at a time.
3. Mark the calculated points in the Hough space.
4. Perform the voting algorithm and determine the lines present in the image.

2 Self-Study Matlab Tasks

2.1 Harris Corner Detector

In many applications in the field of image and video processing it is necessary to describe an image by certain features. Corners are very remarkable and intuitive features of an image that are used very commonly for example in panorama stitching applications.
A well known and widely used technique for the detection of corners in images is the Harris corner detector. Goal of this exercise is to implement the Harris corner detector and test it on an example image.
An outline of the Harris corner detection algorithm is given on page 6-48 of the lecture notes. We will use this outline in this exercise and implement the algorithm step by step.
In the provided file *Harris_Corner_Detector.m* you find a rough outline of the framework we want to use and which you should complete in the following exercise.

1. Therefore, firstly compute the local x and y derivatives of the image by
   $s_x = s[x, y] - s[x - 1, y]$ and $s_y = s[x, y] - s[x, y - 1]$
2. Compute products of local derivatives at every pixel
3. Compute sums of weighted products of derivatives at each pixel. Therefore, you may use the `filter2()` function.
4. Define at each pixel (x,y) Harris structure matrix M as denoted on page 6-45.
5. Compute cornerness c[x,y] at each pixel (x,y) as shown on page 6-46.
6. Apply nonmax suppression.
   (a) Therefore, define a matrix cornerCandidates by applying a threshold on c.
   (b) For the nonmax suppression check a 3x3 neighbourhood and save the maximum value in the result matrix.
7. Return the x and y coordinates of the detected corners.
8. Show the image and plot the detected corners.

1 Pre-Recorded Tasks

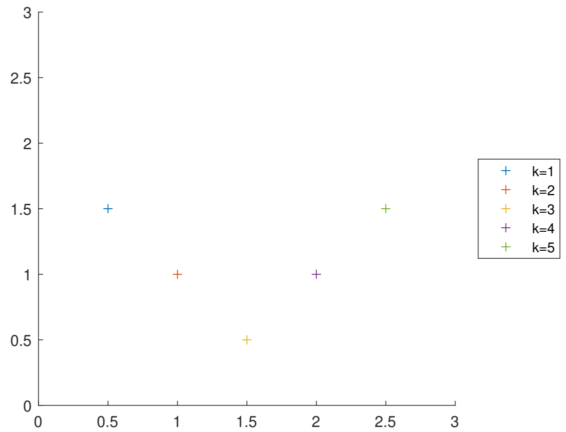
1.1 Hough Transfrom

As the result of an edge detection algorithm we receive the following five points in the space of a Cartesian coordinate system:

$$x(k) = \begin{bmatrix} 0.5 & 1.0 & 1.5 & 2.0 & 2.5 \end{bmatrix}$$

$$y(k) = \begin{bmatrix} 1.5 & 1.0 & 0.5 & 1.0 & 1.5 \end{bmatrix}$$

1. Plot the detected edges in a diagram.



2. We want to use the Hough Transformation to identify straight lines in the image. For our calculations we use the Hesse normal form following page (6-35) of the script. In order to decrease the amount of calculations we can assume that a line goes through at least two edge points at a time.

   For $k = 1$ and $k = 2$:

$$\begin{aligned}
x(1)\cos(\theta_{12}) + y(1)\sin(\theta_{12}) &= x(2)\cos(\theta_{12}) + y(2)\sin(\theta_{12}) \\
0.5\cos(\theta_{12}) + 1.5\sin(\theta_{12}) &= 1\cos(\theta_{12}) + 1\sin(\theta_{12}) \\
\cos(\theta_{12}) &= \sin(\theta_{12}) \\
\theta_{12_1} &= \frac{\pi}{4} \\
\theta_{12_2} &= -\frac{3}{4}\pi \\
\rho_{12_1} = 0.5\cos\left(\frac{\pi}{4}\right) + 1.5\sin\left(\frac{\pi}{4}\right) &= \sqrt{2} \approx 1.41 \\
\rho_{12_2} = 0.5\cos\left(-\frac{3}{4}\pi\right) + 1.5\sin\left(-\frac{3}{4}\pi\right) &= -\sqrt{2} \approx -1.41
\end{aligned}$$

   Only $\theta_{12_1} = \frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

For $k = 1$ and $k = 3$:

$$\begin{aligned}
x(1)\cos(\theta_{13}) + y(1)\sin(\theta_{13}) &= x(3)\cos(\theta_{13}) + y(3)\sin(\theta_{13}) \\
0.5\cos(\theta_{13}) + 1.5\sin(\theta_{13}) &= 1.5\cos(\theta_{13}) + 0.5\sin(\theta_{13}) \\
\cos(\theta_{13}) &= \sin(\theta_{13}) \\
\theta_{13_1} &= \frac{\pi}{4} \\
\theta_{13_2} &= -\frac{3}{4}\pi \\
\rho_{13_1} = 0.5\cos\left(\frac{\pi}{4}\right) + 1.5\sin\left(\frac{\pi}{4}\right) &= \sqrt{2} \approx 1.41 \\
\rho_{13_2} = 0.5\cos\left(-\frac{3}{4}\pi\right) + 1.5\sin\left(-\frac{3}{4}\pi\right) &= -\sqrt{2} \approx -1.41
\end{aligned}$$

Only $\theta_{13_1} = \frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

For $k = 1$ and $k = 4$:

$$\begin{aligned}
x(1)\cos(\theta_{14}) + y(1)\sin(\theta_{14}) &= x(4)\cos(\theta_{14}) + y(4)\sin(\theta_{14}) \\
0.5\cos(\theta_{14}) + 1.5\sin(\theta_{14}) &= 2\cos(\theta_{14}) + 1\sin(\theta_{14}) \\
3\cos(\theta_{14}) &= \sin(\theta_{14}) \\
\tan(\theta_{14}) &= 3 \\
\theta_{14_1} &\approx 1.25 \\
\theta_{14_2} &\approx -1.89 \\
\rho_{14_1} = 0.5\cos(1.25) + 1.5\sin(1.25) &\approx 1.58 \\
\rho_{14_2} = 0.5\cos(-1.89) + 1.5\sin(-1.89) &\approx -1.58
\end{aligned}$$

Only $\theta_{14_1} \approx 1.25$ is a valid solution, as negative radii aren't allowed.

For $k = 1$ and $k = 5$:

$$\begin{aligned}
x(1)\cos(\theta_{15}) + y(1)\sin(\theta_{15}) &= x(5)\cos(\theta_{15}) + y(5)\sin(\theta_{15}) \\
0.5\cos(\theta_{15}) + 1.5\sin(\theta_{15}) &= 2.5\cos(\theta_{15}) + 1.5\sin(\theta_{15}) \\
\cos(\theta_{15}) &= 0 \\
\theta_{15_{1,2}} &= \pm\frac{\pi}{2} \\
\rho_{15_{1,2}} = 0.5\cos\left(\pm\frac{\pi}{2}\right) + 1.5\sin\left(\pm\frac{\pi}{2}\right) &= \pm 1.5
\end{aligned}$$

Only $\theta_{15_1} = \frac{\pi}{2}$ is a valid solution, as negative radii aren't allowed.

For $k = 2$ and $k = 3$:

$$
\begin{aligned}
x(2)\cos(\theta_{23}) + y(2)\sin(\theta_{23}) &= x(3)\cos(\theta_{23}) + y(3)\sin(\theta_{23}) \\
1\cos(\theta_{23}) + 1\sin(\theta_{23}) &= 1.5\cos(\theta_{23}) + 0.5\sin(\theta_{23}) \\
\cos(\theta_{23}) &= \sin(\theta_{23}) \\
\theta_{23_1} &= \frac{\pi}{4} \\
\theta_{23_2} &= -\frac{3}{4}\pi \\
\rho_{23_1} = \cos\left(\frac{\pi}{4}\right) + 1.5\sin\left(\frac{\pi}{4}\right) &= \sqrt{2} \approx 1.41 \\
\rho_{23_2} = \cos\left(-\frac{3}{4}\pi\right) + 1.5\sin\left(-\frac{3}{4}\pi\right) &= -\sqrt{2} \approx -1.41
\end{aligned}
$$

Only $\theta_{23_1} = \frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

For $k = 2$ and $k = 4$:

$$
\begin{aligned}
x(2)\cos(\theta_{24}) + y(2)\sin(\theta_{24}) &= x(4)\cos(\theta_{24}) + y(4)\sin(\theta_{24}) \\
1\cos(\theta_{24}) + 1\sin(\theta_{24}) &= 2\cos(\theta_{24}) + 1\sin(\theta_{24}) \\
\cos(\theta_{24}) &= 0 \\
\theta_{24_{1,2}} &= \pm\frac{\pi}{2} \\
\rho_{24_{1,2}} = \cos\left(\pm\frac{\pi}{2}\right) + \sin\left(\pm\frac{\pi}{2}\right) &= \pm 1
\end{aligned}
$$

Only $\theta_{24_1} = \frac{\pi}{2}$ is a valid solution, as negative radii aren't allowed.

For $k = 2$ and $k = 5$:

$$
\begin{aligned}
x(2)\cos(\theta_{25}) + y(2)\sin(\theta_{25}) &= x(5)\cos(\theta_{25}) + y(5)\sin(\theta_{25}) \\
1\cos(\theta_{25}) + 1\sin(\theta_{25}) &= 2.5\cos(\theta_{25}) + 1.5\sin(\theta_{25}) \\
\sin(\theta_{25}) &= -3\cos(\theta_{25}) \\
\tan(\theta_{25}) &= -3 \\
\theta_{25_1} &\approx 1.89 \\
\theta_{25_2} &\approx -1.25 \\
\rho_{25_1} = 2.5\cos(1.89) + 1.5\sin(1.89) &\approx 0.63 \\
\rho_{25_2} = 2.5\cos(-1.25) + 1.5\sin(-1.25) &\approx -0.63
\end{aligned}
$$

Only $\theta_{25_1} \approx 1.89$ is a valid solution, as negative radii aren't allowed.

For $k = 3$ and $k = 4$:

$$
\begin{aligned}
x(3)\cos(\theta_{34}) + y(3)\sin(\theta_{34}) &= x(4)\cos(\theta_{34}) + y(4)\sin(\theta_{34}) \\
1.5\cos(\theta_{34}) + 0.5\sin(\theta_{34}) &= 2\cos(\theta_{34}) + 1\sin(\theta_{34}) \\
\sin(\theta_{34}) &= -\cos(\theta_{34}) \\
\theta_{34_1} &= -\frac{\pi}{4} \\
\theta_{34_2} &= \frac{3}{4}\pi \\
\rho_{34_1} = 1.5\cos\left(-\frac{\pi}{4}\right) + 0.5\sin\left(-\frac{\pi}{4}\right) &= \frac{\sqrt{2}}{2} \approx 0.71 \\
\rho_{34_2} = 1.5\cos\left(\frac{3}{4}\pi\right) + 0.5\sin\left(\frac{3}{4}\pi\right) &= -\frac{\sqrt{2}}{2} \approx -0.71
\end{aligned}
$$

Only $\theta_{34_1} = -\frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

For $k = 3$ and $k = 5$:

$$
\begin{aligned}
x(3)\cos(\theta_{35}) + y(3)\sin(\theta_{35}) &= x(5)\cos(\theta_{35}) + y(5)\sin(\theta_{35}) \\
1.5\cos(\theta_{35}) + 0.5\sin(\theta_{35}) &= 2.5\cos(\theta_{35}) + 1.5\sin(\theta_{35}) \\
\sin(\theta_{35}) &= -\cos(\theta_{35}) \\
\theta_{35_1} &= -\frac{\pi}{4} \\
\theta_{35_2} &= \frac{3}{4}\pi \\
\rho_{35_1} = 1.5\cos\left(-\frac{\pi}{4}\right) + 0.5\sin\left(-\frac{\pi}{4}\right) &= \frac{\sqrt{2}}{2} \approx 0.71 \\
\rho_{35_2} = 1.5\cos\left(\frac{3}{4}\pi\right) + 0.5\sin\left(\frac{3}{4}\pi\right) &= -\frac{\sqrt{2}}{2} \approx -0.71
\end{aligned}
$$

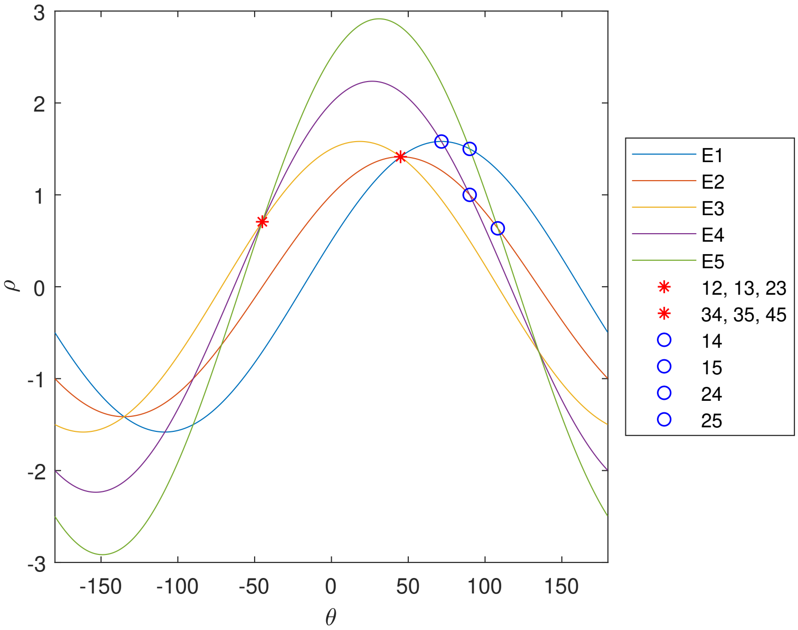
Only $\theta_{35_1} = -\frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

For $k = 4$ and $k = 5$:

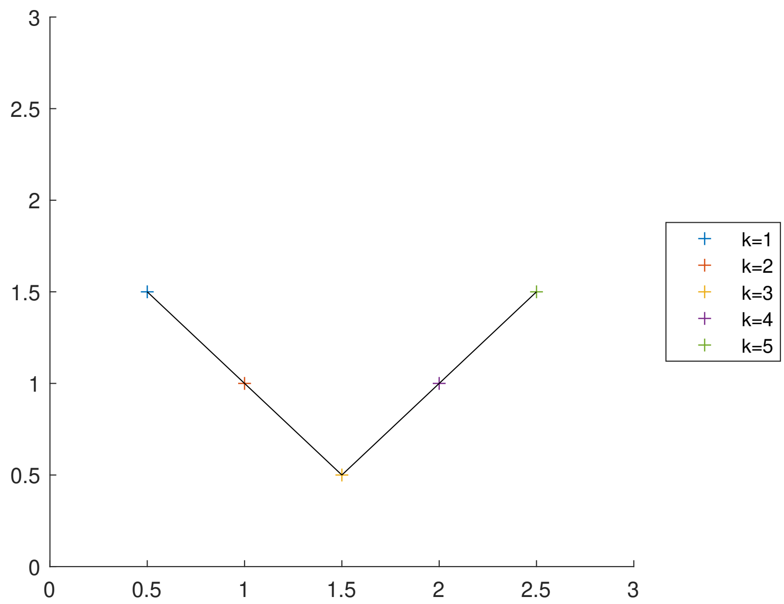
$$
\begin{aligned}
x(4)\cos(\theta_{45}) + y(4)\sin(\theta_{45}) &= x(5)\cos(\theta_{45}) + y(5)\sin(\theta_{45}) \\
2\cos(\theta_{45}) + 1\sin(\theta_{45}) &= 2.5\cos(\theta_{45}) + 1.5\sin(\theta_{45}) \\
\sin(\theta_{45}) &= -\cos(\theta_{45}) \\
\theta_{45_1} &= -\frac{\pi}{4} \\
\theta_{45_2} &= \frac{3}{4}\pi \\
\rho_{45_1} = 2\cos\left(-\frac{\pi}{4}\right) + \sin\left(-\frac{\pi}{4}\right) &= \frac{\sqrt{2}}{2} \approx 0.71 \\
\rho_{45_2} = 2\cos\left(\frac{3}{4}\pi\right) + \sin\left(\frac{3}{4}\pi\right) &= -\frac{\sqrt{2}}{2} \approx -0.71
\end{aligned}
$$

Only $\theta_{45_1} = -\frac{\pi}{4}$ is a valid solution, as negative radii aren't allowed.

3. Mark the calculated points in the Hough space.



4. Perform the voting algorithm and determine the lines present in the image. We receive the most votes for $\theta = \frac{\pi}{4}$ and $\rho = \sqrt{2}$ as well as for $\theta = -\frac{\pi}{4}$ and $\rho = \frac{\sqrt{2}}{2}$. Therefore we receive the following image:



2 Self-Study Matlab Tasks

2.1 Harris Corner Detector

In many applications in the field of image and video processing it is necessary to describe an image by certain features. Corners are very remarkable and intuitive features of an image that are used very commonly for example in panorama stitching applications.
A well known and widely used technique for the detection of corners in images is the Harris corner detector. Goal of this exercise is to implement the Harris corner detector and test it on an example image.
An outline of the Harris corner detection algorithm is given on page 6-48 of the lecture notes. We will use this outline in this exercise and implement the algorithm step by step.
In the provided file *Harris_Corner_Detector.m* you find a rough outline of the framework we want to use and which you should complete in the following exercise.

1. Therefore, firstly compute the local x and y derivatives of the image by
   $s_x = s[x, y] - s[x - 1, y]$ and $s_y = s[x, y] - s[x, y - 1]$
2. Compute products of local derivatives at every pixel
3. Compute sums of weighted products of derivatives at each pixel. Therefore, you may use the `filter2()` function.
4. Define at each pixel (x,y) Harris structure matrix M as denoted on page 6-45.
5. Compute cornerness c[x,y] at each pixel (x,y) as shown on page 6-46.
6. Apply nonmax suppression.
   (a) Therefore, define a matrix cornerCandidates by applying a threshold on c.
   (b) For the nonmax suppression check a 3x3 neighbourhood and save the maximum value in the result matrix.
7. Return the x and y coordinates of the detected corners.
8. Show the image and plot the detected corners.