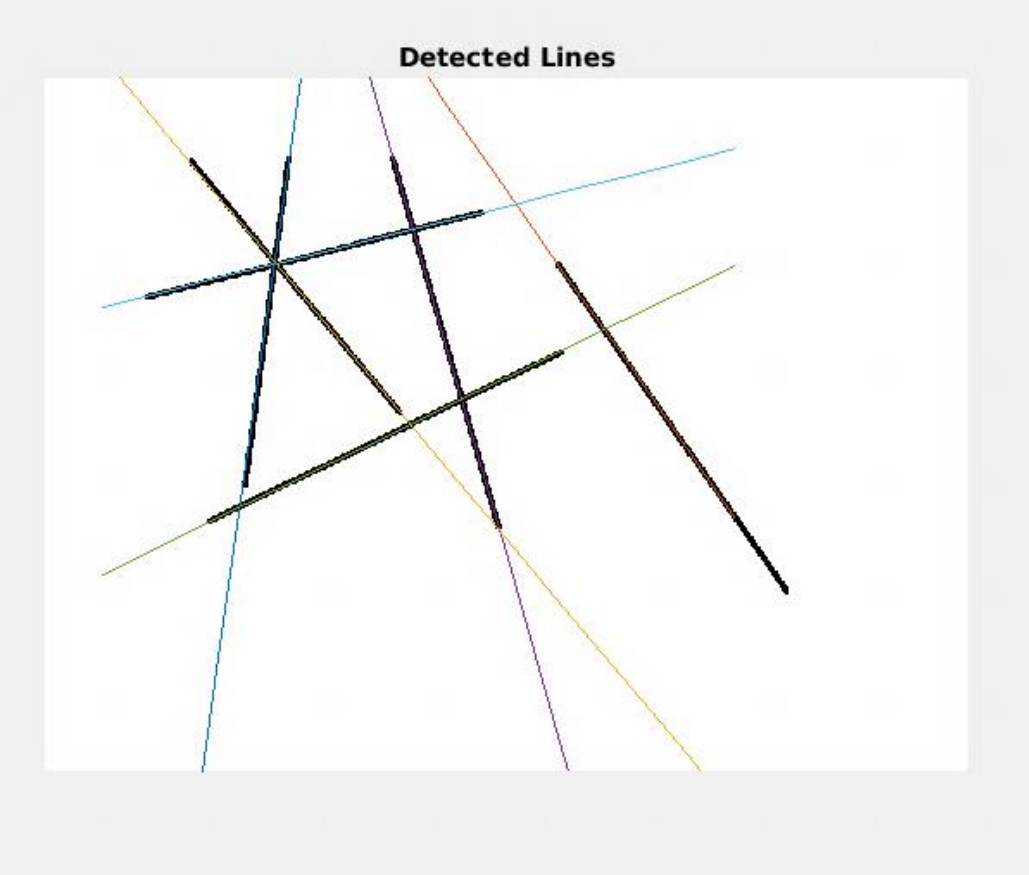
Code Walkthrough and Results discussion

Detecting Lines:

1. Following are the steps followed to detect lines in an image
2. Convert the input image to gray scale and skeletonize it using built-in bwmorph() function.
3. Find the gradient direction using Sobel edge operator and imgradient() built-in function.
4. Convert the input image to hough space and find the peaks to get line parameters as follows:
    a. Initialize an accumulator array to zeros for all values of β and Θ.
    b. XY plane coordinates are converted to βΘ plane coordinates by setting the range of β and Θ.
    c. Iterate through all the edge pixel positions and possible values of Θ to get β. Increment the value in accumulator array for every β and Θ.
    d. Find the maximum peak above certain intensity threshold and set the neighboring area to zeros.
    e. Do step d until you find required number of peaks.
    f. Plot the lines using the parameters obtained above which look as shown in image below



The image above shows the plotted colored output lines along with the input white lines. In the hough transform we take each edge pixel and consider all the possible lines that pass through that point and cast votes accordingly. The cell with the maximum votes gives the line parameters. Same concept was used here to detect the lines. We can see they overlap with the input lines as shown above.

Execution Steps:

Run the script Lines.m. It calls the line_hough.m function internally and computes the parameters, plots the lines as shown above.

Following are the parameters obtained in ρΘ plane for the lines given

```
rho=272  theta=-35
rho=271  theta=-16
rho=205  theta=76
rho=220  theta=8
rho=50  theta=-40
rho=409  theta=64
```

Detecting Ellipses:

Following are the steps followed to detect axis aligned ellipses in an image.

1. Convert the input image to gray scale and find the edges using built-in edge() function and Sobel operator.
2. Find the gradient direction using Sobel edge operator and imgradient() built-in function.
3. Convert the input image to hough space and find the peaks to get ellipse parameters as follows:
    a. Initialize an accumulator array to zeros for all values of x,y,a and b.
    b. Iterate through all the edge pixel positions and possible values of a,b to compute the ellipse center x,y. Increment the value in accumulator array for every x,y,a,b.
    c. Find the maximum peak above certain intensity threshold and set the neighboring area to zeros.
    d. Do step d until you find required number of peaks.
    e. Plot the lines ellipses the parameters obtained above which look as shown in image below

The image above shows the plotted colored output ellipses along with the input white white. In the hough transform we take each edge pixel and consider all the possible ellipses that pass through that point and cast votes accordingly. The cell with the maximum votes gives the ellipse parameters. Same concept was used here to detect the axis aligned ellipses. We can see that the output ellipses overlap with the input ellipses as shown above.

Execution Steps:

Run the script Ellipses.m. It calls the ellipse_hough.m function internally and computes the parameters, plots the ellipses as shown above.

Following are the parameters obtained for the given ellipses

x=369, y=122, a=86, b=30
x=158, y=270, a=71, b=68
x=179, y=67, a=86, b=30
x=255, y=173, a=156, b=29

Detection of vertical ellipses:

The code can detect even vertical ellipses as shown below. Please uncomment line 17 in Ellipses.m to find vertical ellipses as well. Use the following image as input.