Student: Chandan Chandel

Student Number: 250914472

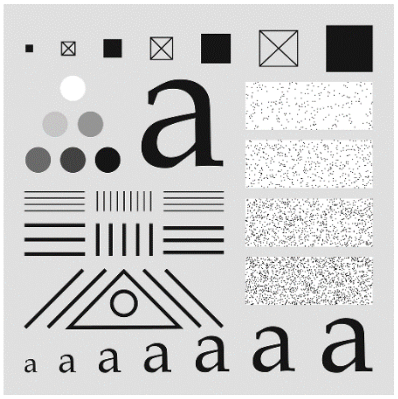
Course: Advanced Image Processing

Professor: Dr. Yimin Yang

Assignment 4 – Kirsch Compass Kernels for Edge Detection

Summary: I completed Edge Detection using Kirsch Compass Kernels in python and used the resulting gradients to build a magnitude image and an angle image. I then compared the Kirsch magnitude image to two different version of the Sobel Magnitude image (one version using imfilter and the other edge).

Data Description: I used the testpattern512.tfif which is shown below.



Magnitude image generation process:

1. Get all eight direction compass kernels.
2. Using a for loop to cycle through each of the kernels, convolve the image with the current kernel, then take the absolute value of each resulting gradient and append to a list of gradients.

Please note that each gradient is of the same size

3. I cycled through each pixel (i,j) of each gradient and found the largest pixel value. The largest pixel value was assigned the i,j location of the magnitude image.

4. I then found the largest value in the magnitude image(V) and then thresholded it using the following formula `magnitude_image > T*V`. If true the pixel is set to V else to 0.

* Please note that T is given by the user*

Angle image generation process:

1. Create an angle conversion dictionary and get all eight direction compass kernels.
2. Using a for loop to cycle through each of the kernels, convolve the image with the current kernel, then append the gradient to a list of gradients.

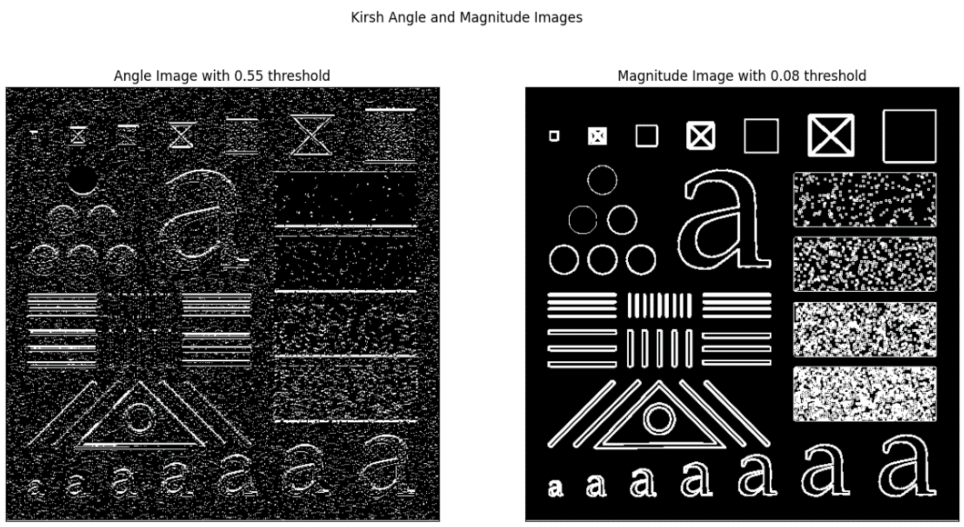
Please note that each gradient is of the same size

3. I cycled through each pixel (i,j) of each gradient and found the largest pixel value and recorded which direction kernel it came from. The largest pixel value's direction was then converted to the angle in the angle conversion dictionary and was assigned to the i,j location of the magnitude image.

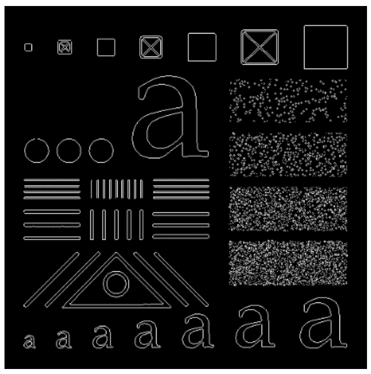
4. I then found the largest value in the angle image(V) and then thresholded it using the following formula ` angle_image >  round(T*V/45) * 45`. If true the pixel is set to V else to 0.

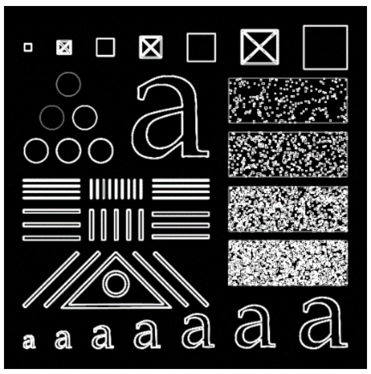
* Please note that T is given by the user*

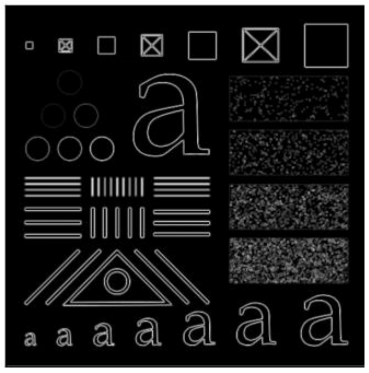
Output from Kirsch Angle & Magnitude Generation Process:


Kirsh Angle and Magnitude Images
Angle Image with 0.55 threshold
Magnitude Image with 0.08 threshold

Comparing the magnitude images from the Sobel kernel and the Kirsch kernel:


Sobel Magnitude Edge


Sobel Magnitude ImFilter

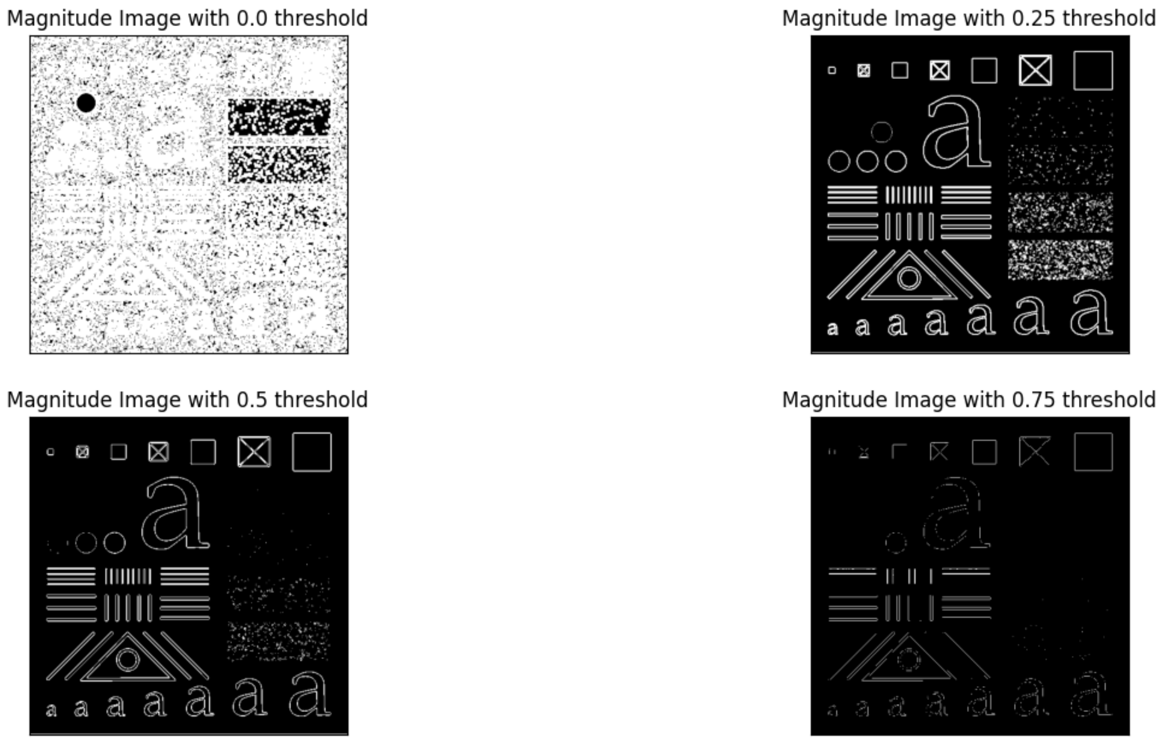

Kirsch Magnitude (no thresholding)

The Sobel Edge struggled a lot with the thinner circles whereas every line was visible with the Sobel ImFilter. The Kirsch Magnitude image with no threshold does show all the lines but some are very faint.

I asked Chat-GPT why Sobel Edge performed less well on the thin circles than Sobel ImFilter and it replied by saying that the edge detection applies some thresholding while ImFilter does not. This made a lot of sense to me as thresholding can lose some information while highlighting other information.

This also made me curious to understand the effects of thresholding on my magnitude image with my Kirsch Magnitude Images, so I wrote a function to compare a few of the side by side, and the results are shown below. The obvious conclusion is that as thresholding becomes more and more extreme, more and more edges are lost. This shows what we found originally as the discrepancy between the Sobel Edge Image and the Sobel ImFilter Image, can be explained by thresholding.

Magnitude Images at Different Thresholds


Magnitude Image with 0.0 threshold
Magnitude Image with 0.25 threshold
Magnitude Image with 0.5 threshold
Magnitude Image with 0.75 threshold