Assignment1

Jingyuan Wang

Question1

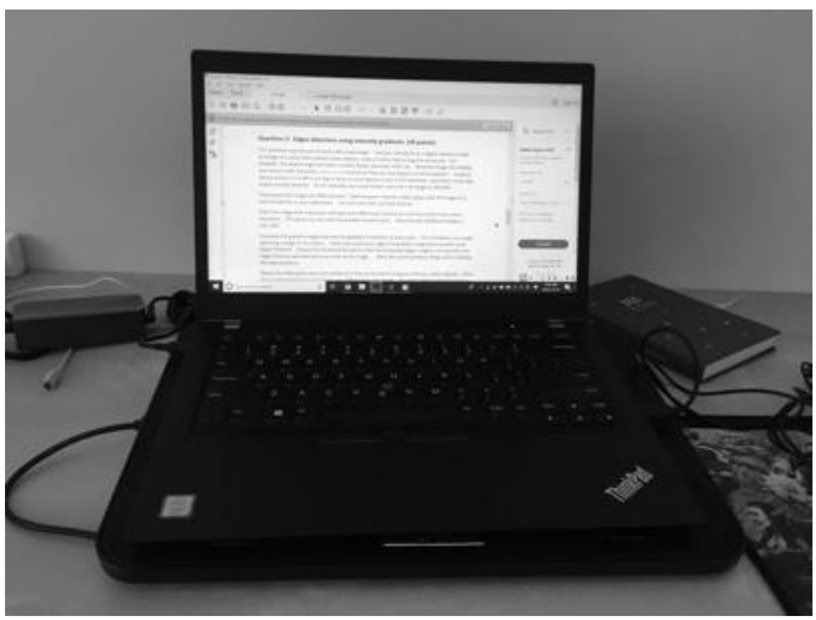
Please see codes and comments in the 2 required files:

a) 'make2DGaussian.m'
b) 'myConv2.m'

In myConv2.m, although the professor said we can assume N to be odd, I still made it work for both odd and even N so it can basically get the same result with conv2().

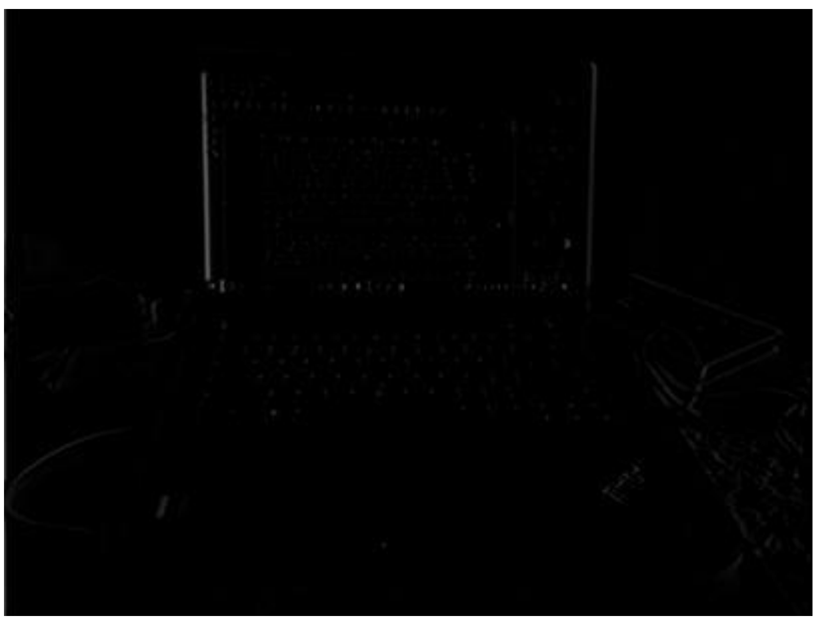
Question2

a) Matlab codes and comments: 'Question2.m'
b) Original image: 'myimage2.png'
c) Gray scale JPG image: 'image_gray.jpg'



d) The pair of filtered images(with Gaussian and take local differences):

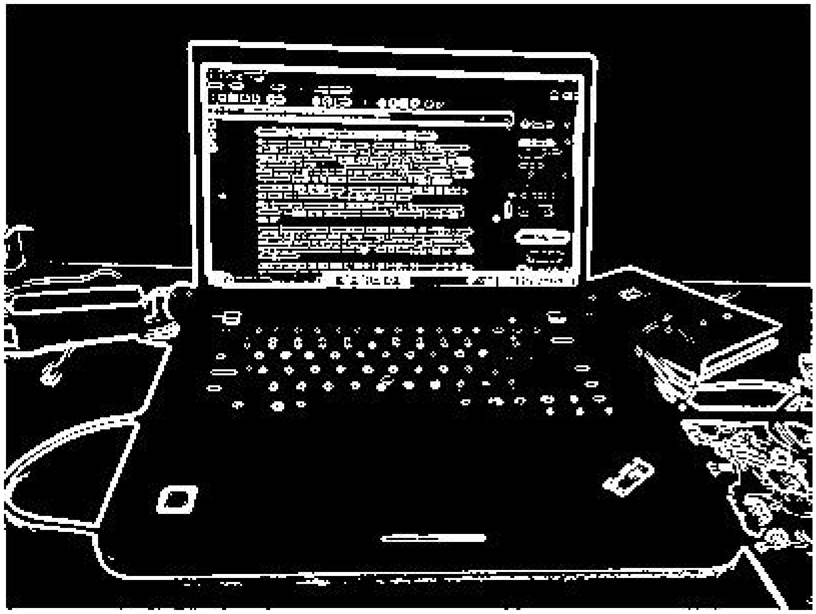
Local differences in x direction



Local differences in y direction

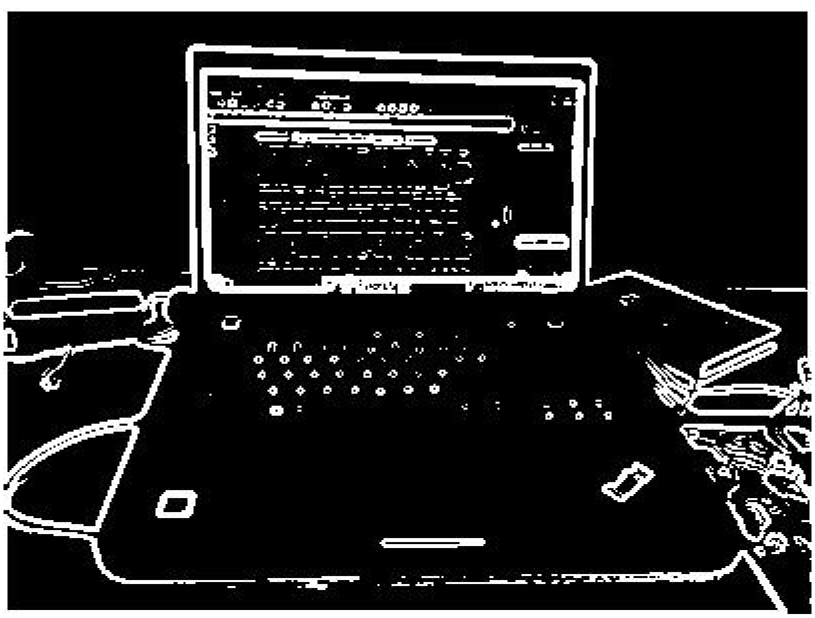
e) Compute the image gradient magnitude and orientation with matlab function grandient(), which returns the gradient magnitude in x and y directions. With variables [Fx,Fy], calculate $\sqrt{{Fx_{i,j}}^2 + {Fy_{i,j}}^2}$ as magnitude and $\tan^{-1}\frac{Fy_{i,j}}{Fx_{i,j}}$ as orientation radians.

After several attempts, I chose 10 to be the threshold of gradient magnitude and made the binary image, which is as below:

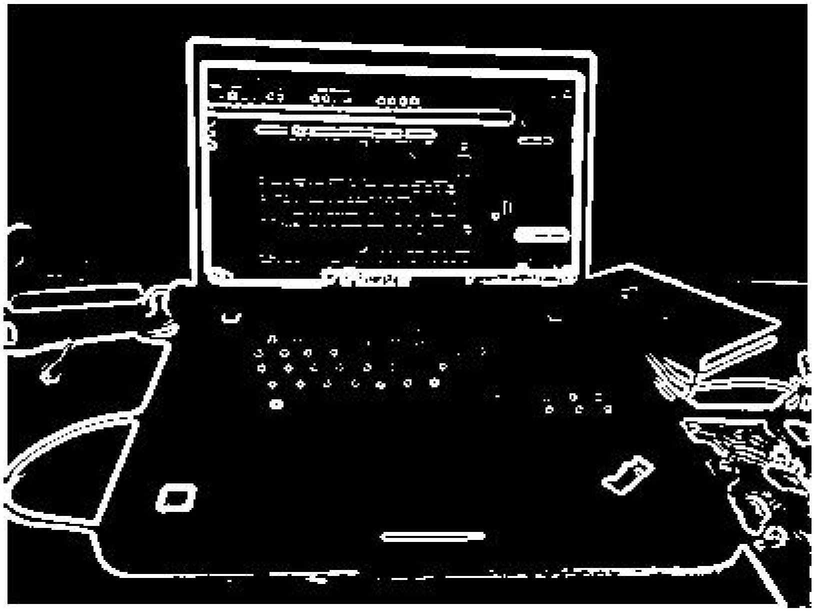


Edge map with σ = 0.5

f) In previous tasks, I was taking a normal value of σ to be 0.5, so I made a loop to compute matrices and pictures with σ = 1 and 2



Edge map with σ = 1



Edge map with σ = 2

If we compare these 3 binary edge images, it can be easily found out that the first image, with the smallest σ, includes most details, like the all the words on the laptop screen, letter on the keyboard as well as patterns on the mouse pad, which are more or less ignored by the other two filters. In other word, filtered image with smaller σ detects most edges, since the horizontal line of the desk is shown in the first pic but are missing in the others.

However, pics with larger σ can leave out more noises, like the lower edge of the laptop and the words on screen and mouse pad patterns as well if we count it as meaningless noises. Also, larger σ gets the edge thicker and smoother because the Gaussian filter spare the intensities of the edge to the adjacent pixels.

Question3

a) My implementation of LOG filter: ‘make2DLOG.m’

Used formula $\mathrm{h}(n_1, n_2) = \frac{(n_1^2+n_2^2-2\sigma^2)h_g(n_1,n_2)}{\sigma^4 \sum_{n_1}\sum_{n_2} h_g}$, and normalized the sum of kernel elements to be 0.

b) Code and comments in file: ‘Question3.m’

c) Original image:’myimage3.jpg’

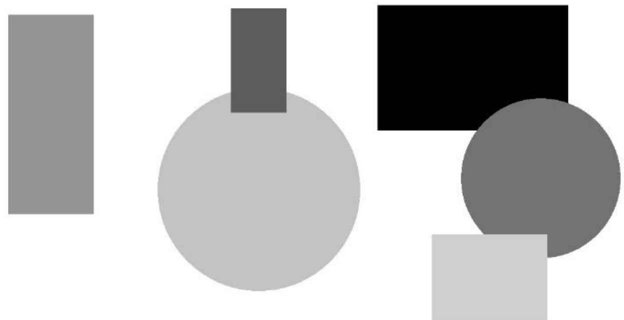
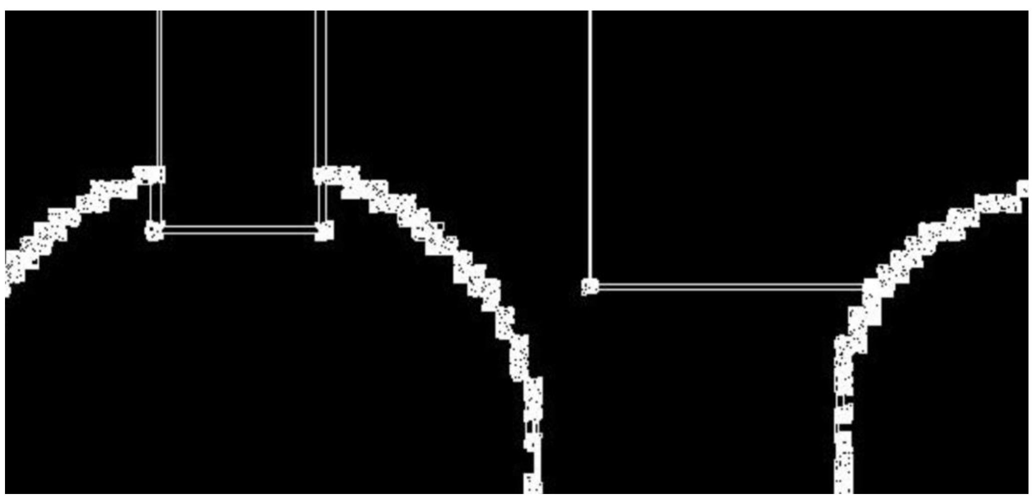
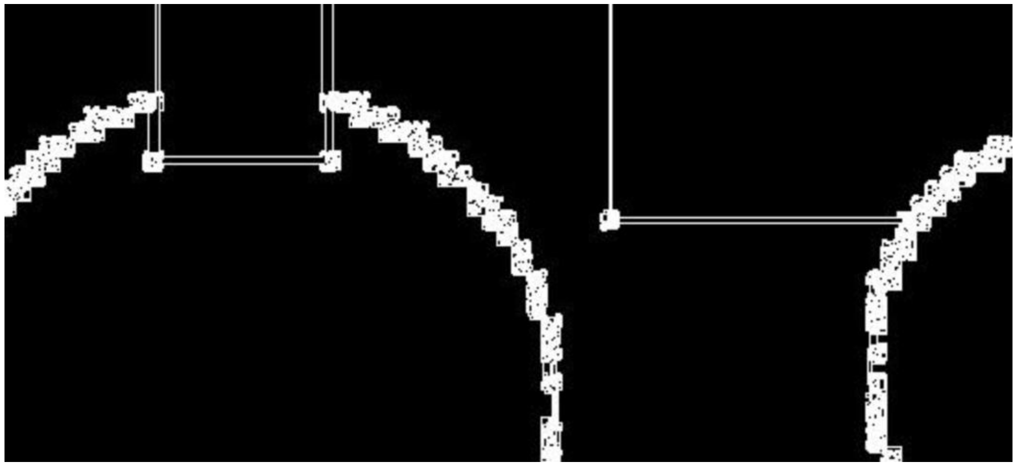


Image with rectangles and circles

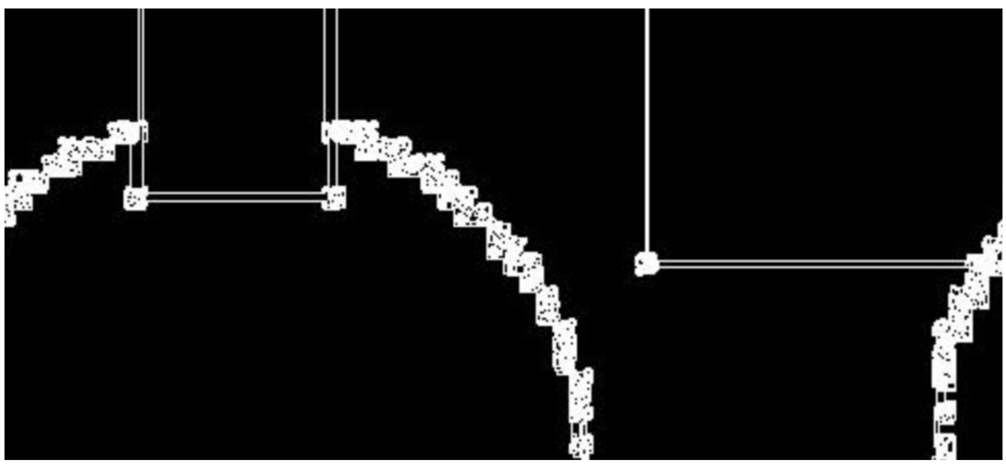
d) After we make a binary image out of the zero-crossing points, there appears several sharp and straight edge on edges of the rectangle and thick zigzagged ones on the disks. When we increase the value of σ, the edges and joints it forms remain the same structures but become a little thicker since filtering with large σ will spreads out the intensities and gets smoother derivatives and second derivatives and lower peaks, so that two or three more pixels will distinguish themselves to be zero-crossing points.
Moreover, large σ will result in more non-zero-crossing points mixed in the thick edge formed by zero-crossing points. As the edges blurred, some small spots may share the similar value of intensities, which leads to small black dots in the binary image.



Zero crossing points with $\sigma = 0.5$



Zero crossing points with $\sigma = 1$

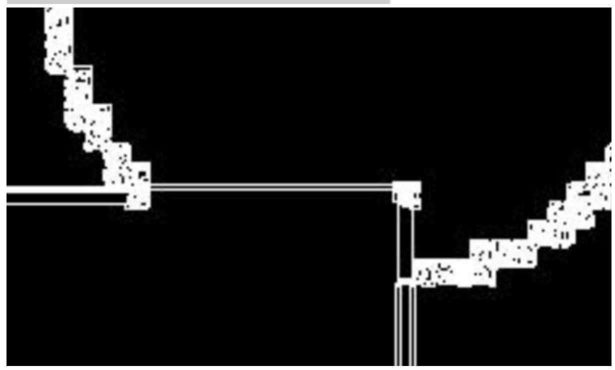


Zero crossing points with $\sigma = 4$

Partial binary images of zero crossing points are showed above. The full images look identical in this small scale, so I prefer to skip them in this pdf. Please refer to the whole output image files which I put in the same folder. Here are the file names with the σ values in the prefixes.

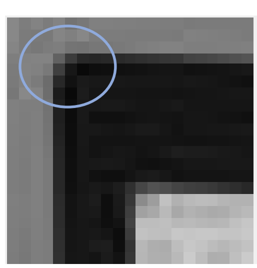
‘0.5_zerocrossing.jpg’
‘1_zerocrossing.jpg’
‘2_zerocrossing.jpg’
‘4_zerocrossing.jpg’

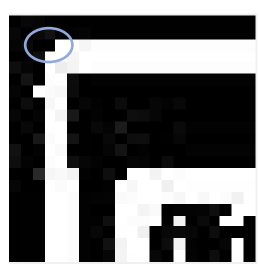
e) Below are the original image and zero-crossing image of several ‘T junctions’. It is obvious that at T junctions and corners, zero-crossing points are close to each other and form a small square. That’s because such points has more zero-crossing directions. If we see the horizontal line segment in the middle as a isolated straight line, it can be found that the zero-crossing points are extremely uniform in this area. And closer, fewer lines will form in areas with larger differences of intensity.



Question4

a) Please see code in: ‘Question4.m’
I chopped a 20*20 pixels image from Question2, which is a corner of the laptop screen. As shown in the images below, the original image is composed of a black corner, a gray background on the left and upper region, and a bright screen rectangle in the lower right part. In the edge map, 2 group of edges at right angles was white because in these parts, the intensity changes sharply. Also some of the white dots in the lower right corner of the second picture represents noises which cannot be classified as edges.

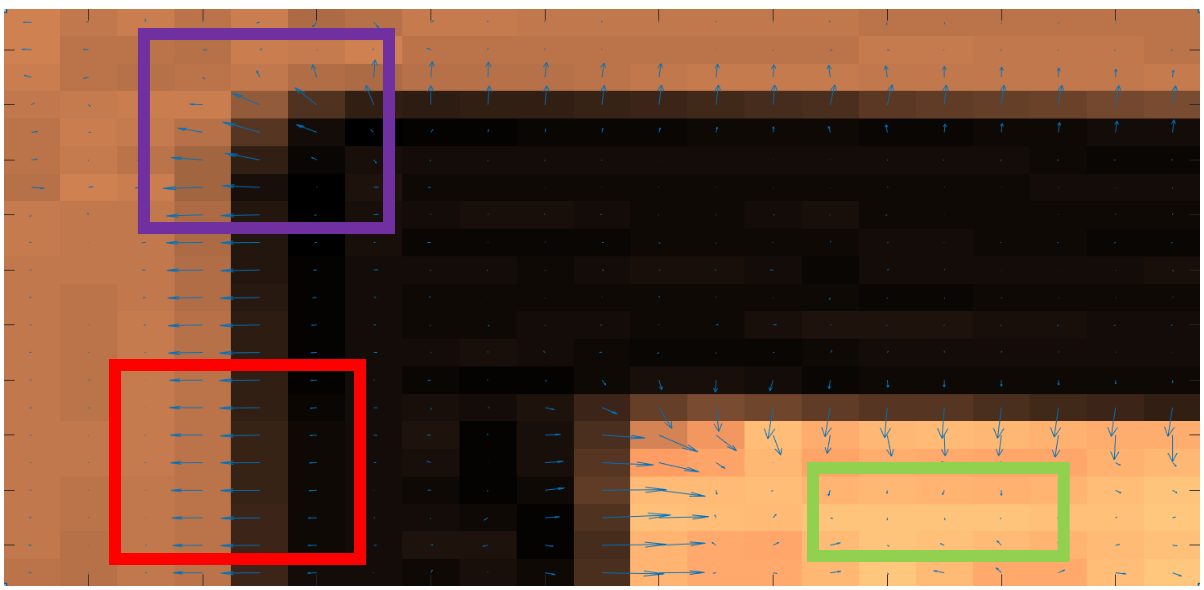




Another interesting consequence is that both two corners in the image have lost a pixel on the upper left corner. It is partly due to the intensity dispersion by the gaussian filter. In the border of 2 different intensity regions, like the original intensities on the left picture, may take up to 6 pixels of high intensity to do the gaussian convolution while the corner(which may be on the exact top of the left pic), has only 4 high intensity pixels. So after the convolution, the corner pixel has lower intensity than ones on the right or lower side of it.

border-edge corner-edge

b) I used imagesc() function to change the original image matrix into a colored picture and overlay it with the quiver() plot.

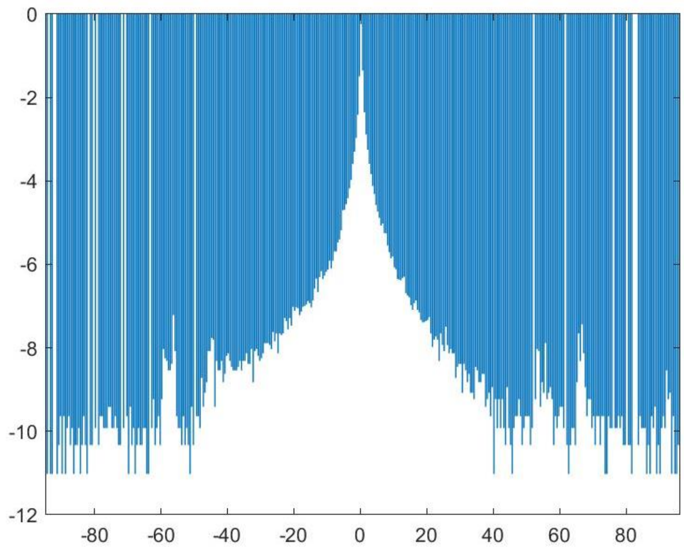


At intensity edges, take the region in the red frame for example, the intensity gradient starts from a small value, increase to a relatively large value, and then decrease to small value along the direction of the arrows. The parallel gradients along the direction of straight line will present similar values and same directions and vary at the same speed. Similarly, in the purple frame, gradient intensities raises and then decreases and diverge to different directions because there are unequal intensity differences in both x and y direction.

However, in a region of flat intensities, such as the green frame, gradients value are normally small and have different orientations which is because the intensities in this region have similar values and noises make them slightly vary.

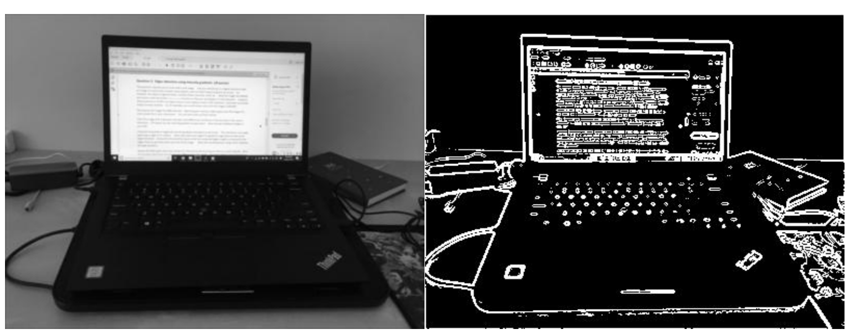
Question5

a) Code in: 'Question5.m'
b) In this part, I used histcounts() to count the frequency of local differences in x direction and made a distribution histogram with bar-chart in matlab. Both normalization and logarithm was applied and below are the results I got.



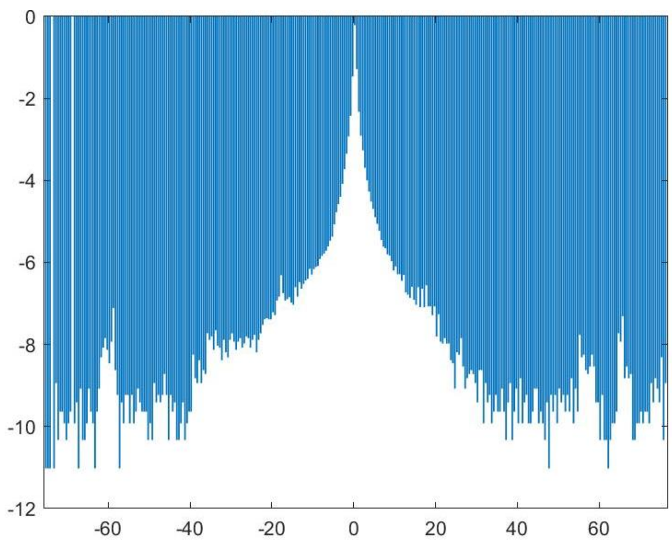
Histogram of partial derivates of x with $\sigma = 0.5$

With enough sample with substantial pixels, we get a Gaussian-like distribution plot. However, the data values of these histograms has a smaller range than of gaussian distribution which has a domain of all real numbers. Besides, the frequencies of the histogram do not decrease continuously as the absolute value of data increases but has increased a little in the area of long tail and presented an irregular distribution.

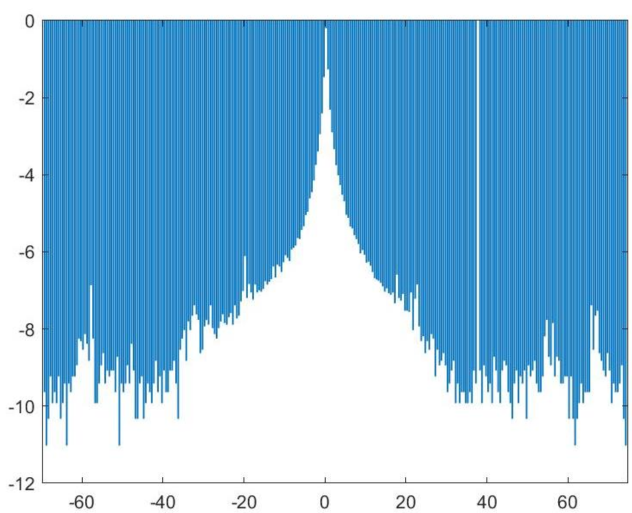


Gaussian filtered image and edge map

The reasons this difference occur are due to the nonuniform distribution of image intensities. Firstly, the gradient of pixels are in fact bounded in a specific region. Think of a pixel with its right pixel to be completely white and left one to be completely black. Then the local difference of this pixel in x direction would be the largest, which has the value of 0.5*(255-0). With Gaussian smoothing the intensity, the max value will be even smaller. Another important reason is that such pictures has a relatively large area of edges. As you can see, the picture has the two largest part of bright wall & desk background as well as the black laptop body, (which makes the frequencies around the zero point in the histogram to be extremely large). The next obvious feature of the image is edges with all those white pixels in the edge map meaning that the local differences of these pixels are quite large. So I think the gradient are not distributed evenly like a complete noise image and the quite big scale of large gradients (representing edges) has increased the frequencies on the end of two tails.



Histogram of partial derivates of x with $\sigma = 1$



Histogram of partial derivates of x with $\sigma = 2$

When σ becomes bigger, the range of the ‘x’ values becomes smaller. As discussed previously, the maximum and minimum x value of this distribution depend on the largest gradient of the filtered image. When we apply the Gaussian filter with a larger σ, the intensity of a pixel will be influenced by more adjacent pixels and the sharp difference of the edge region is smoothed to a smaller one.