ID student: 0780828

Name: Alisher Mukashev

Class: Digital Image Processing

Introduction

This overview aims to introduce the report for the project 1 in the course 5051 Digital Image Processing. This report covers the solution for three assigned tasks. These tasks are described in the document ***reportProject1.pdf*** published on the website e3new for the course. This report has four separate sections which were written as standalone. Each section describes its assigned task and presents a theoretical framework for its solution.

Section 1

The task: Apply the same strategy of Figures 3.57 to the image file, *dew on roses (noisy).tif*. Plot all the intermediate (in-process) images as illustrated in Figure 3.57 (a)-(h).

Figure 1-1.

Background

This section refers to an Image Enhancement problem. For such purpose given image *dew on roses (noisy).tif* (Figure 1-1).

The mentioned strategy is described in the section 3.8 Combining Spatial Enhancement Methods of the book Digital Image Processing of Rafael Gonzalez et al. This strategy details seven consecutive steps which are listed as following:

1. Laplacian filter application.

Laplacian Operator is a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask, Eq. (1-1). In this mask we have two further classifications one is Positive Laplacian Operator and other is Negative Laplacian Operator.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{1-1}$$

In our task we consider Negative Laplacian Operator $k_{laplacian}(x,y)$:

−1	−1	−1
−1	8	−1
−1	−1	−1

Figure 1-2.

After applying negative Laplacian operator by using Eq. (1-2) we will get the Laplacian of input image *f(x,y)* (Figure 1-6). We can notice that this figure has such view because we are dealing with noisy image.

$$g(x,y) = f(x,y) * k_{laplacian}(x,y) \tag{1-2}$$

2. Sharpened Image using Laplacian filter

If we apply negative Laplacian operator then we have to add the resultant image onto original image to get the sharpened image (Figure 1-7). Remember that in our case *c=1*:

$$f(x,y)_{Sharpened} = f(x,y) + c * g(x,y) \tag{1-3}$$

3. Sobel gradient of image.

In Sobel operator (Figure 1-3) we have allotted more weight to the pixel intensities around the edges. Figure 1-8 shows the Sobel gradient of the original image, computed using Eq. (1-6). As expected, edges are much more dominant in this image than in the Laplacian image.

−1	−2	−1
0	0	0
1	2	1

a

−1	0	1
−2	0	2
−1	0	1

b

Figure 1-3, a-sobelX, b-sobelY

$$g(x,y)_{sobelX} = f(x,y) * sobelX \tag{1-4}$$

$$g(x,y)_{sobelY} = f(x,y) * sobelY \tag{1-5}$$

$$f(x,y)_{sobel} = |g(x,y)_{sobelX}| + |g(x,y)_{sobelY}| \tag{1-6}$$

4. Sobel image smoothing

The smoothed gradient image in Figure 1-9 was obtained by using a box filter of size 5x5 (K_{box}) Eq. (1-7):

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Figure 1-4

$$f(x,y)_{Msobel} = f(x,y)_{sobel} * K_{box} \tag{1-7}$$

5. Combined application of Laplacian and Sobel filter

The product of the Laplacian in Eq. (1-2) and smoothed-gradient image in Eq. (1-7) is shown in Figure 1-10.

6. Sharpened image by Laplacian and Sobel filter

Adding the product of the Laplacian and smoothed-gradient image to the original resulted in the sharpened image shown in Figure 1-11.

7. Power-law (Gamma) Transformation

The general form of Power law (Gamma) transformation function is

$$s = c * r^{\gamma} \tag{1-8}$$

Where, s and r are the output and input pixel values, respectively and c and γ are the positive constants. Like log transformation, power law curves with $\gamma < 1$ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher input values. Similarly, for $\gamma > 1$, we get the opposite result.

If images are not gamma-encoded, they allocate too many bits for the bright tones that humans cannot differentiate and too few bits for the dark tones. So, by gamma encoding, we remove this artifact.

Images which are not properly corrected can look either bleached out, or too dark.

Let's verify that $\gamma < 1$ produces images that are brighter (Figure 1-12) while $\gamma > 1$ results in images that are darker (Figure 1-13) than intended.

Algorithm, Flow-chart

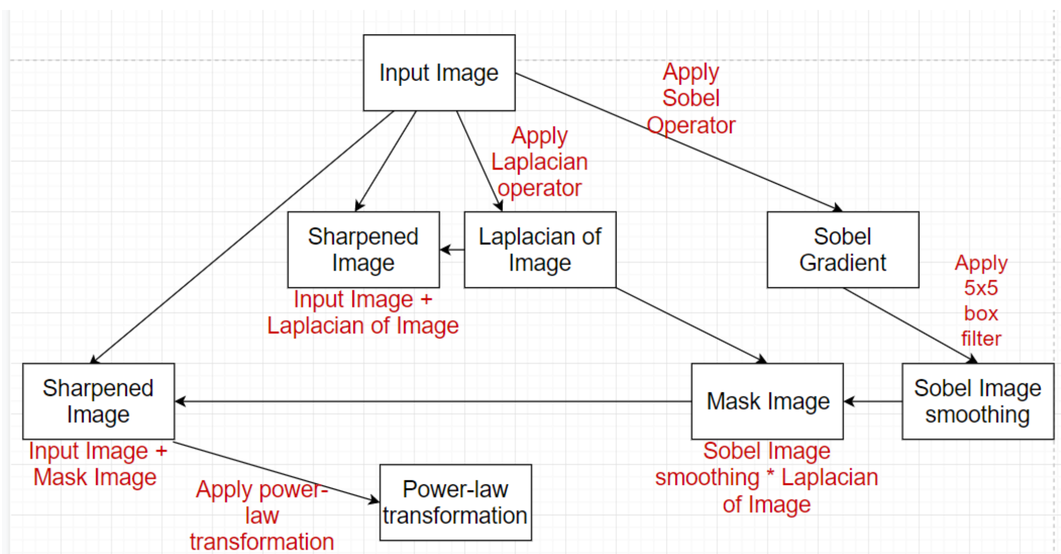

Figure 1-5.

Results

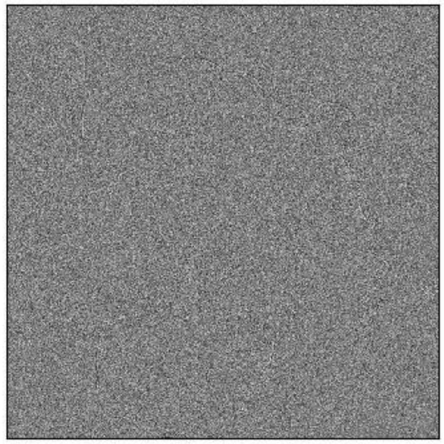

Figure 1-6.

Figure 1-7.

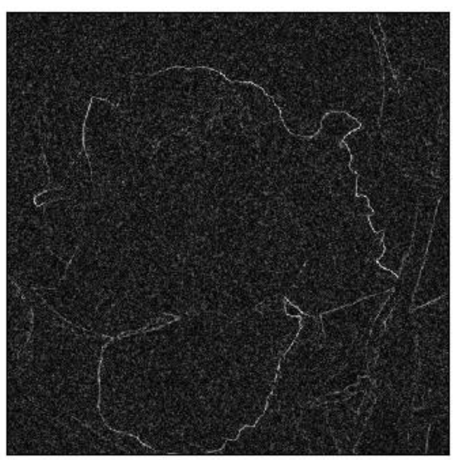

Figure 1-8.

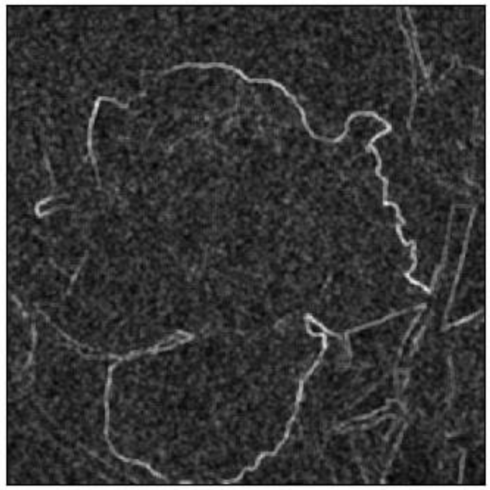

Figure 1-9.

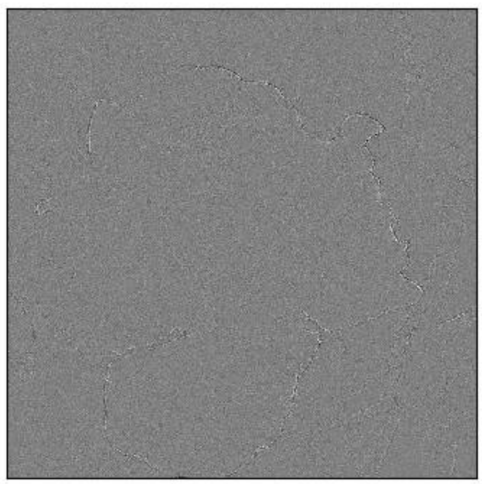

Figure 1-10.

Figure 1-11.

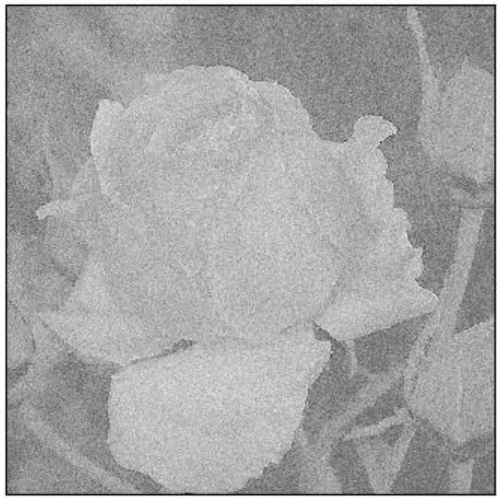

Figure 1-12.

Figure 1-13

Discussions

In this section we studied combining spatial enhancement methods. As we can see we can’t use this method for noisy image which we have in this task. The results are not so good as we expected. The noise can’t allow us highlight edges. I would recommend try to use filters before for getting enhance image in the end.

Section 2

The task: Consider the centered DFT for *dew on roses (noisy).tif* and *tulips irises.tif*, (i) resynthesize the images using the DFT coefficients inside the circular region with radius=30 pixels (based on the original image size), plot the resulted images; (ii) similar to problem (i), however, use the DFT coefficients outside the circular region.

Figure 2-1

Figure 2-2

Background

This section refers to Filtering in Frequency Domain problem. For such purpose given image *dew on roses (noisy).tif* (Figure 2-1) and *tulips irises.tif* (Figure 2-2).

Steps to implement this task:

1. Given an input image *f(x, y)* of size *MxN* obtain the padding parameters *P* and *Q*. Typically, we select *P=2*M* and *Q=2*N*. This process is called zero padding. I used two kind of padding (Figure 2-4, 2-5), but still got the same results.

2. Compute the DFT of input image, *f(x,y)*, Eq. (2-1) after zero padding (1024x1024).

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) * e^{-i*2*\pi*(u*\frac{x}{M}+v*\frac{y}{N})} \qquad (2\text{-}1)$$

3. Translate to center of the frequency rectangle, Eq. (2-2), Figure 2-6:

$$F'(u,v) = F(u - \frac{M}{2}, v - \frac{N}{2}) \qquad (2\text{-}2)$$

4. We have two problems: consider obtained DFT coefficients inside the circular region = 30 pixels (based on the original image size, it means that we will use value = 60 pixels, because of zero padding and increasing original image to 1024x1024); consider obtained DFT coefficients outside the same circular region.

5. Use Eq. (2-3) to get circle. Each pixel makes equal to zero when *D(u,v)>=60* for the first problem (Eq. (2-4), Figure 2-7, Figure 2-9) and when *D(u,v)<=60* for the second problem (Eq. (2-5), Figure 2-8, Figure 2-10).

$$D(u,v) = [\left(u - \frac{P}{2}\right)^2 + \left(u - \frac{Q}{2}\right)^2]^{1/2} \qquad (2\text{-}3)$$

$$G(u,v) = \begin{cases} F'(u,v), & D(u,v) \geq 60 \\ 0, & otherwise \end{cases} \qquad (2\text{-}4)$$

$$G(u,v) = \begin{cases} F'(u,v), & D(u,v) \leq 60 \\ 0, & otherwise \end{cases} \qquad (2\text{-}5)$$

6. Compute the IDFT for both problems by using Eq. (2-6).

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) * e^{-i*2*\pi*(u*\frac{x}{M}+v*\frac{y}{N})} \qquad (2\text{-}6)$$

7. Crop obtained image (512x512).

Algorithm, Flow chart

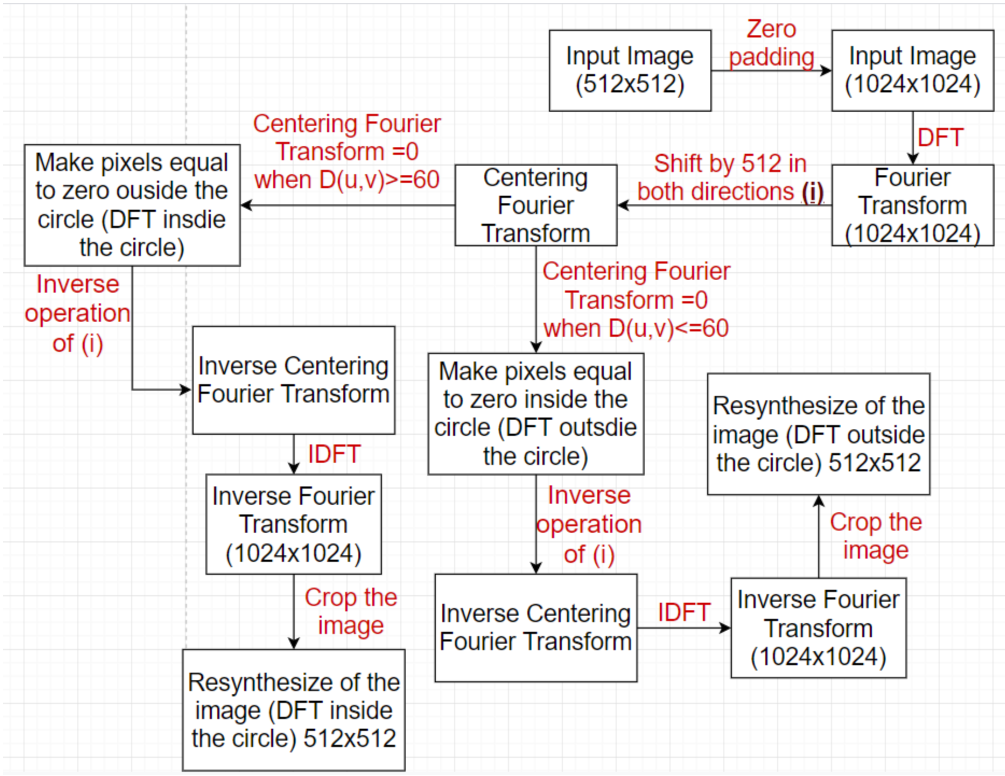

Figure 2-3

Results

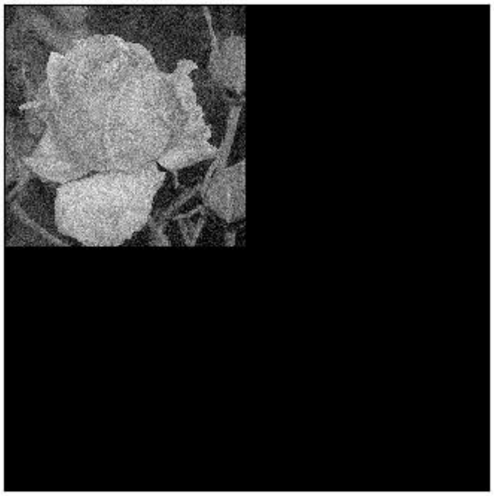

Figure 2-4

Figure 2-5

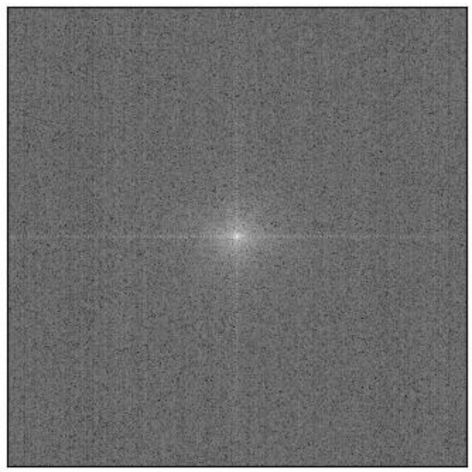

Figure 2-6

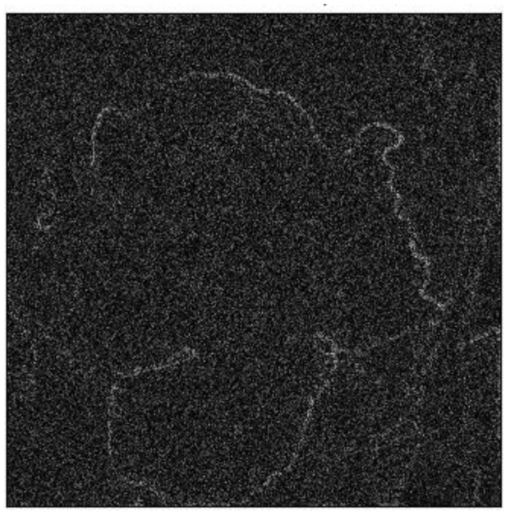

Figure 2-7

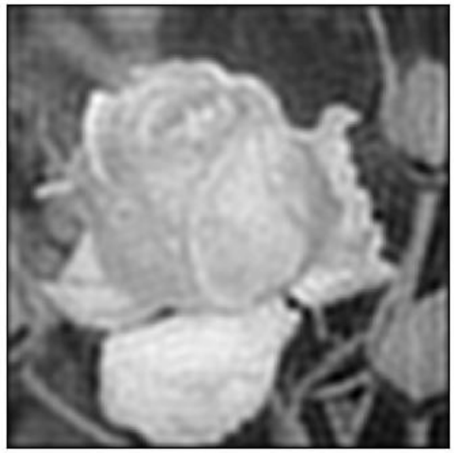

Figure 2-8

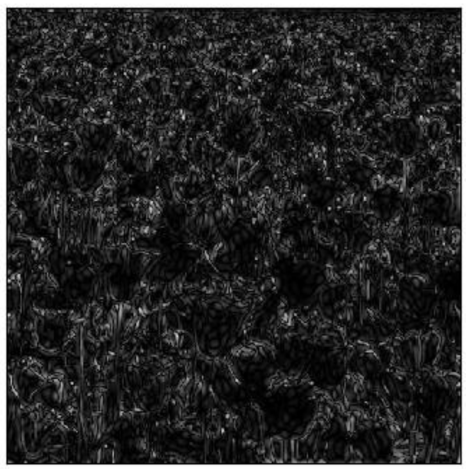

Figure 2-9

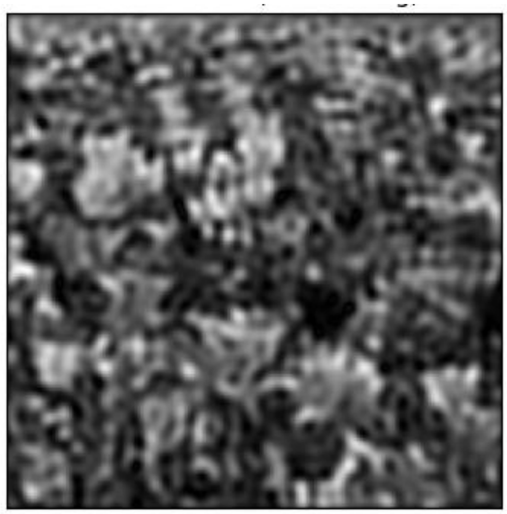

Figure 2-10

Discussions

We can notice that in the first problem (using the DFT coefficients inside the circular region) we are dealing with Highpass filter and in the second problem it is Lowpass filter.

Section 3.a

The task: Determine the possible noise model and model parameters for the *noise in dew on roses (noisy).tif*. Determine an appropriate method to reduce the noise and plot the reconstructed image.

Figure 3-1

Background

This section refers to Estimating Noise Parameters problem. For such purpose given image *dew on roses (noisy).tif* (Figure 3-1).

When only images already generated by a sensor are available, frequently it is possible to estimate the parameters of the PDF from small patches of reasonably constant background intensity. After getting this we can compute Histograms (Figure 3-3 (mask is 100:200, 100:200), 3-4 (mask is 200:300, 200:300)). As we can see it is Gaussian Noise model. The PDF of a Gaussian random variable, *z*, is given by using Eq. (3-1):

$$p(z) = \frac{1}{\sqrt{2*\pi*\sigma}} * e^{-\frac{(z-\bar{z})^2}{2*\sigma^2}} \tag{3-1}$$

To get model parameters we can simply use equation for mean (Eq. (3-2)) and variance (Eq. (3-4)):

$$m = \sum_{k=0}^{L-1} z_k * p(z_k) \tag{3-2}$$

$$p(z_k) = \frac{n_k}{MN} \tag{3-3}$$

Where n_k is the number of times that intensity z_k occurs in the image and MN is the total numbers of pixels.

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 * p(z_k) \tag{3-4}$$

In the table you can see obtained results:

Parameter\Figure	Figure 3-3	Figure 3-4
m	158	167
σ^2	822	672
σ	29	26

After that we need to restore the image. I used Gaussian filter because we are dealing with Gaussian nose model. The results shown on Figure 3-5.

Algorithm, Flow chart

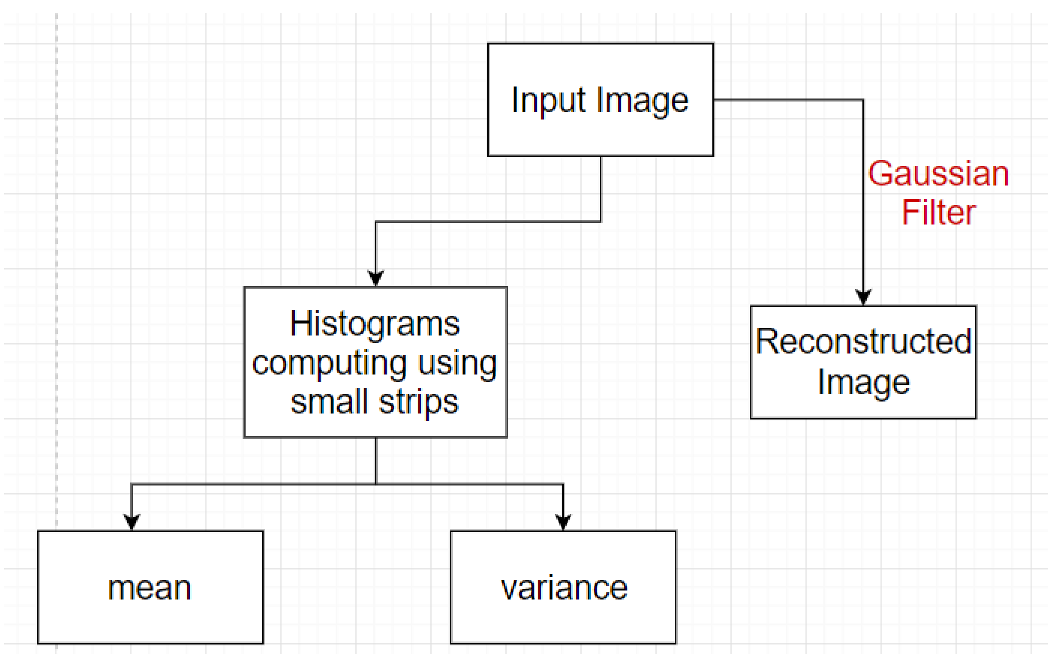

Figure 3-2

Results

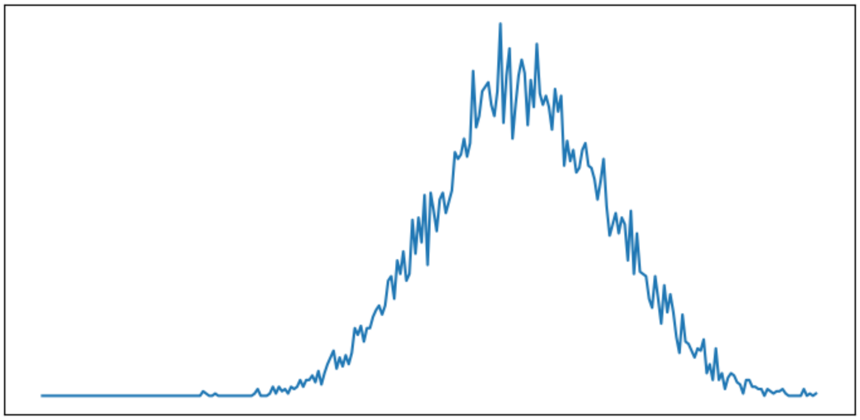

Figure 3-3

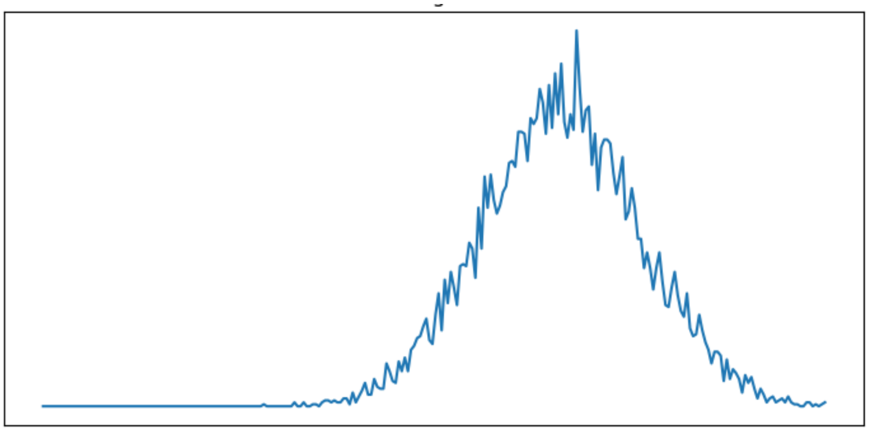

Figure 3-4

Figure 3-5

Discussions

We used simple method of choosing appropriate filter for image.

Section 3.b

The task: Estimate the possible degradation function *H(u,v)* [hint: motion blurring] and determine the model parameters for the degraded image *dew on roses (blurred).tif*. Construct and plot the restored image using the *H(u,v)* obtained.

Figure 3-6

Background

This section refers to Estimation the Degradation Function problem. For such purpose given image *dew on roses (blurred).tif* (Figure 3-6).

To restore image, $\hat{F}(u,v)$, we can use Inverse Filtering:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \tag{3-5}$$

Where *G(u,v)* – degraded image, which we have as input image; *H(u,v)* –degradation function, which we can get by using motion blurring. Shortly about motion blurring:

$$H(u,v) = \frac{T}{\pi*(u*a+v*b)} * \sin[\pi * (u * a + v * b)] * e^{-i*\pi*(u*a+v*b)} \tag{3-6}$$

We can see that model parameters here are *a* and *b*. We can change these parameters and try to get better results. After generating function H(u,v) we use equation to get original image. To select appropriate *a* and *b* we use spectral characteristic of *G(u,v)* and *H(u,v)* in certain *v*, trying to make them equal (Figure 3-8, 3-9).

If the degradation function has zero or very small values, then the ratio *G(u,v)/H(u,v)* would not work. One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin. Thus, by limiting the analysis to frequencies near the origin, we reduce the probability of encountering zero values. Result with cut off outside a radius of 90 is shown in Figure 3-10

Algorithm, Flow chart

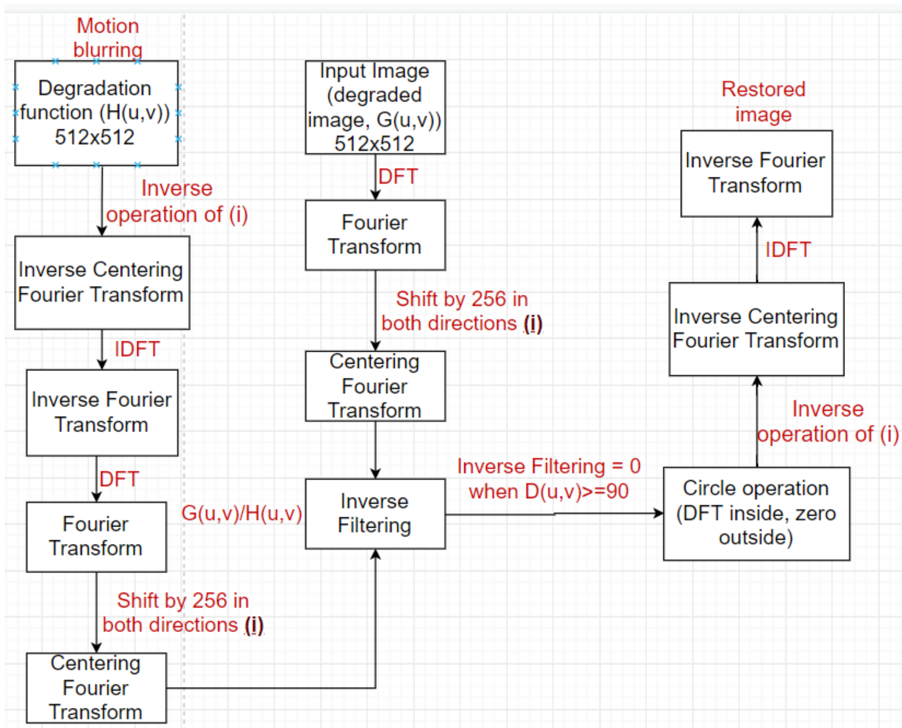

Figure 3-7

Results

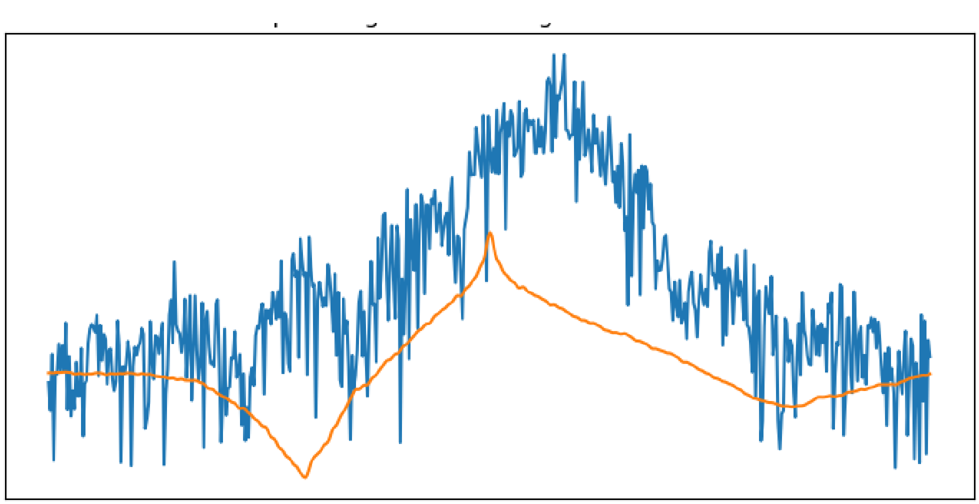

Figure 3-8

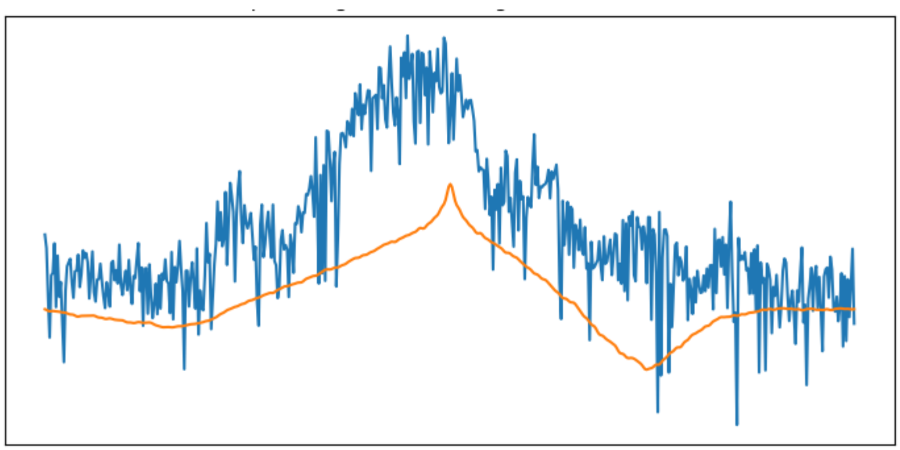

Figure 3-9

Figure 3-10