ID Student: A062689
Name: Bolívar Enrique Solarte
Class: Image Processing
Topic: Project 1

Introduction

This overview aims to introduce the report for the project 1, which was assigned on 4th April 2018 in the course 5055 Image Processing. This report covers the solution for three assigned tasks. These tasks are described in the document ***project 1 2018.pdf*** published on the website ***e3new*** for the course. This report has three separate sections which were written as standalone. Each section describes its assigned task and presents a theoretical framework for its solution. Furthermore, the used algorithms and additional information related to its implementation are also presented.

1. Section I

Assigned task: Apply the same strategy of Figures 3.43 to strawberry grey blurred-noise and wisteria vine grey blurred-noise and plot all the intermediate (in-process) images as illustrated in Figure 3.43(a)-(h).

a) strawberry grey blurred-noise.tif

b) wisteria vine grey blurred-noise.tif.

Figure 1-1 Input Images

Background

This section refers to an Image Enhancement problem. For such purpose, two images are given ***strawberry grey blurred-noise.tif*** and ***wisteria vine grey blurred-noise.tif***. These figure are introduced in the Figure 1-1.

The mentioned strategy (hereinafter referred as strategy) is described in the section 3.7 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
2. Sharpened Image using Laplacian filter
3. Sobel gradient application.
4. Sobel image smoothing
5. Combined application of Laplacian and Sobel filter
6. Sharpened image by Laplacian and Sobel filter
7. Histogram Transformation

The idea behind of the Laplacian application, steps 1 and 2, is to enhance the fine details of the image. But, since the Laplacian is a second derivative operator, it also highlights the noise. In that manner, a combined application with a Sobel filter, which is a gradient operator, allow us to get a sharpened image with an enhancement of fine details.

In such a way the first step is applying Laplacian filter to the original image. The equation (1-1) shows the convolution between a kernel $k_{laplacian}(x,y)$ to the original image $f(x,y)$. This Laplacian implementation consider isotropic results in increments of 45°

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

$$k_{laplacian}(x,y) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{1-2}$$

Since the function $f(x,y)$ represents the original image which it is expected to be noised, then the noise would also amplified in the Laplacian image $g(x,y)$. A solution for that is a blurring filter, like Gaussian or median filter before the Laplacian application, but a blurring filter could delete important features. Nevertheless, the results for the Laplacian filter application to both blurring image and original one are shown in Figure 1-2 and Figure 1-3. It is important to mention; the proposed strategy does not consider a blurring filter before to the Laplacian application, instead, it considers a combined approach using Sobel filter to highlight fine details and deal with the noise problem.

In order to get the a sharpened image $f(x,y)_{sharpened}$ using the Laplacian image $g(x,y)$ the equation (1-3) is applied. Is important to note the constant c can be set to $c = 1$ or $c = -1$ regarding the used kernel. Thus, for the kernel $k_{laplacian}(x,y)$ described in the equation (1-2) $c = -1$. The Figure 1-4 shows the sharpened image $f(x,y)_{sharpened}$

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

So far it was obtained the sharpened image by Laplacian filter application. The next step is to obtain a sharpened image from a Sobel mask, steps 3 and 4. This filter is also sensible to noise and fine details but in lower magnitude than Laplacian filter does. Furthermore, it has a strong average response in areas of significant transition. For that reason, a combined application of Sobel and Laplacian can lead to enhance fine details and reduce the noise

(noise smoothing) in the image. In order to compute the Sobel application, the following equations and kernels are applied to the original image.

$$g(x,y)_{sobel\,X} = f(x,y) \star k_X(x,y) \tag{1-4}$$

$$g(x,y)_{sobel\,Y} = f(x,y) \star k_Y(x,y) \tag{1-5}$$

$$g(x,y)_{sobel} = |g(x,y)_{sobel\,X}| + |g(x,y)_{sobel\,Y}| \tag{1-6}$$

$$k_X(x,y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{1-7}$$

$$k_Y(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1-8}$$

Is important to mention the equation (1-6) is an approximation for the derivative operator of first order which still preserves the relative change in intensity for each dimension. The $g(x,y)_{sobel}$ is shown in the Figure 1-5.

Once the Sobel filter is implemented, it is necessary to apply a median filter to reduces the noise in the $g(x,y)_{sobel}$ image. In such way the combined application of the Laplacian filter and blurred Sobel mask can attenuate such noise and still highlighting the fine details from the Laplacian application. The implementation for such combined application is described by the following equations.

$$g(x,y)_{M_{Sobel}} = g(x,y)_{sobel} \star k_M(x,y) \tag{1-9}$$

$$k_M(x,y) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{1-10}$$

$$g(x,y)_{Smoothed-Sharpened} = g(x,y)_{M_{Sobel}} \cdot f(x,y)_{sharpened} \tag{1-11}$$

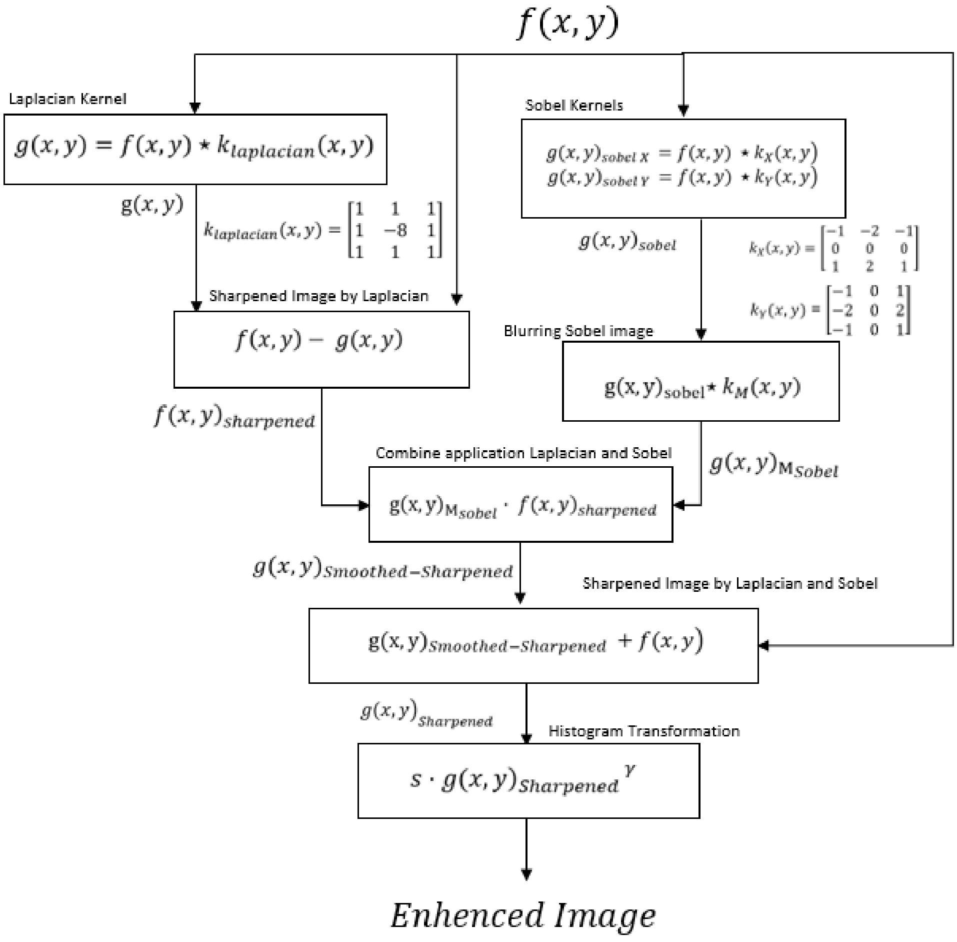
$$g(x,y)_{Sharpened} = g(x,y)_{Smoothed-Sharpened} + f(x,y) \tag{1-12}$$

The image $g(x,y)_{Smoothed-Sharpened}$ is the combined result of Laplacian and Sobel filters application, where the Sobel filter was before blurred by a median kernel $k_M(x,y)$. In the Figure 1-7 is possible to appreciate the dominance of the strong edges and reduction of visible noise for the image $g(x,y)_{Smoothed-Sharpened}$, which is in itself the objective behind of this strategy. The result function $g(x,y)_{Sharpened}$ represents the sum of the original image $f(x,y)$ and the sharpened image $g(x,y)_{Smoothed-Sharpened}$ from the combined application of Laplacian and Sobel filter. This image is presented in Figure 1-8.

The last step for our strategy is to increase the dynamic range of the sharpened image $g(x,y)_{Sharpened}$. For that, a power-law gamma transformation is used. Thus, the equation (1-13) details its implementation for such histogram transformation. The result image is shown in Figure 1-9 and

$$g(x,y)_{Enhancement} = s \cdot g(x,y)_{Sharpened}{}^{\gamma} \tag{1-13}$$

Algorithm



Results. ***strawberry grey blurred-noise image***

Figure 1-2 Laplacian Image

Figure 1-3 Laplacian Image from Blurred original

Figure 1-4 Sharpened Image by Laplacian filter

Figure 1-5 Sobel Image

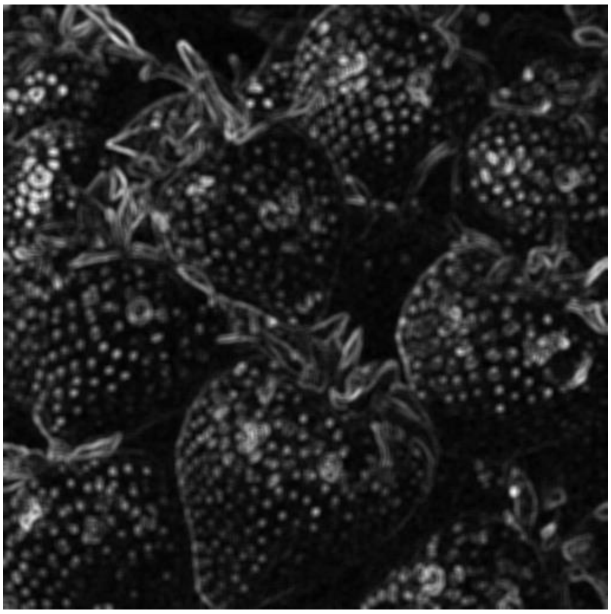


Figure 1-6 Blurred Sobel Image

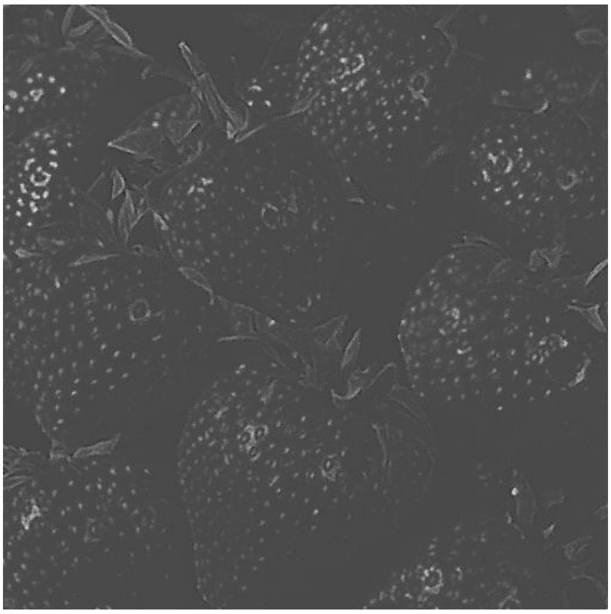


Figure 1-7 Combine Application of Laplacian and Sobel filters

Figure 1-8 Sharpened Image by Laplacian and Sobel filters

Figure 1-9 Histogram Transformation

Results. *wisteria vine grey blurred-noise image*

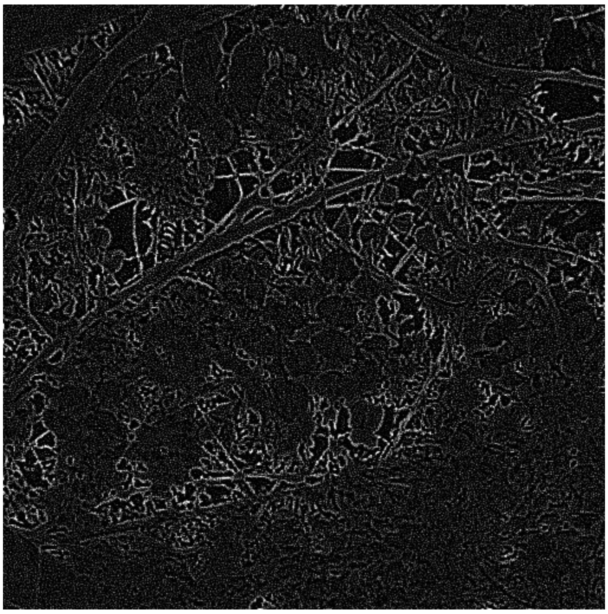


Figure 1-10 Laplacian Image

Figure 1-11 Sharpened Image by Laplacian filter

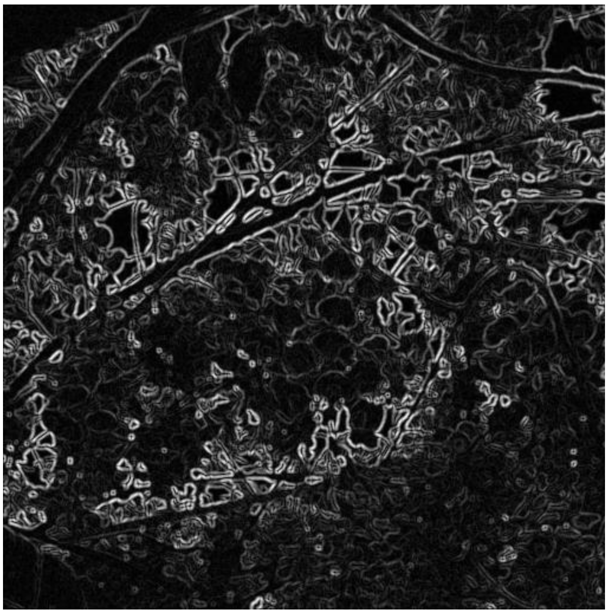


Figure 1-12 Sobel Image magnitude

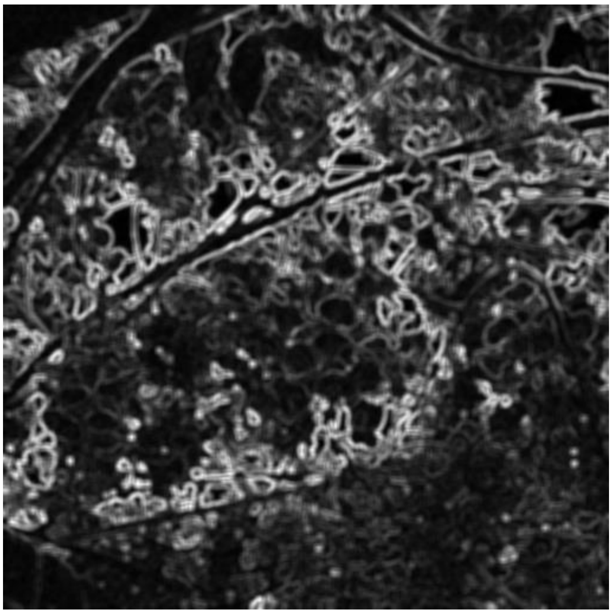


Figure 1-13 Blurred Sobel Image

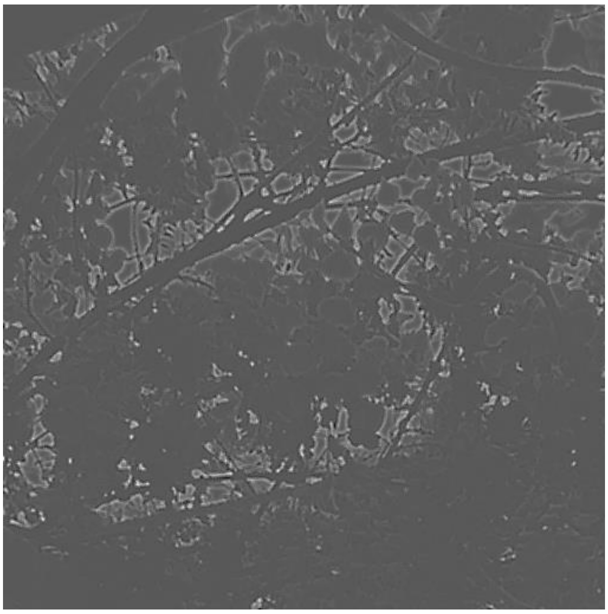


Figure 1-14 Combine application of Laplacian and Sobel filters

Figure 1-15 Sharpened Image by Laplacian and Sobel filter application

Figure 1-16 Histogram Transformation

Discussions.

Figure 1-17 Original Image for strawberry grey blurred-noise image

Figure 1-18 Enhanced Image for strawberry grey blurred-noise image

The idea behind of an enhancement image is to highlight the features of such image. The strategy implemented in this section considers a combine application between a Laplacian and Sobel filter. Both algorithms enhance areas of significant transition, but Laplacian is more sensible to noise than Sobel. In Such way a combine implementation of these filters allow us uses the advantage from both filters.

It is important to note; the original images were not blurred before to the Laplacian filter implementation. Even though this algorithm also boosts the noise in the image, the Sobel filter allow us to deal and enhance the response of such Laplacian filter implementation.

Figure 1-19 Original Image for wisteria vine grey blurred-noise image

Figure 1-20 Enhanced Image for wisteria vine grey blurred-noise image

2. Section II

Assigned task: According to the DFT property of Laplacian, it appears we may implement Laplacian operation by designing a digital filter with frequency response $H(u, v) = k(u^2 + v^2)$ where k is a scaling factor to make the magnitude of $H(u, v)$in the range [0, 1]. Use this frequency-domain scheme to find the Laplacian images for ***strawberry grey blurred*** and ***wisteria vine grey blurred***, and compare the results of applying Laplacian mask (Chapter 3) to both images. Plot (1) the frequency response of $H(u, v)$, (2) Laplacian images using respectively filtering method and mask operation and (3) their DFT magnitude spectra.

Figure 2-1 Image $f(x, y)$ strawberry grey blurred

Figure 2-2 Image $f(x, y)$ wisteria vine grey blurred

Background

This section treats filtering in frequency domain and its relation in spatial domain. But before going into the solution for the assigned task let's define some important concepts.

First of all, it is possible to define the frequency representation of an image applying the Discrete Fourier Transform (hereinafter referred as DFT). This representation is given by the equation (2-1), which defines the function $F(u,v)$ in frequency domain based on a function $f(x,y)$ in spatial domain. Furthermore, the inverse transformation from frequency domain to spatial domain is named inverse Discrete Fourier Transform (hereinafter referred as iDFT). This iDFT allows us to define a $f(x,y)$ based on its representation $F(u,v)$ in frequency domain (equation (2-2)).

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \cdot e^{-j2\pi\frac{ux}{M}} \cdot e^{-j2\pi\frac{vy}{N}} \tag{2-1}$$

$$f(x,y) = \frac{1}{M \cdot N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \cdot e^{j2\pi\frac{ux}{M}} \cdot e^{j2\pi\frac{vy}{N}} \tag{2-2}$$

Once the DFT and iDFT have been defined, it is possible to introduce one of the properties for DFT- iDFT which is going to help us in solving the assigned task for this section. This property is the convolution definition in spatial and frequency domain.

The convolution in spatial domain allows us to implement a kernel filter based on a certain mask (matrix $h(x,y)$). This convolution can be defined by the equation (2-3). Moreover, this convolution is also defined as the product $F(u,v) \cdot H(u,v)$, which in itself is the frequency domain representation for $f(x,y)$ and $h(x,y)$. Thus, we can formalize this property by the equations (2-4) and (2-5)

$$g(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) \cdot h(x-m, y-n) = f(x,y) \star h(x,y) \tag{2-3}$$

$$f(x,y) \star h(x,y) \leftrightarrow DFT \leftrightarrow F(u,v) \cdot H(u,v) \tag{2-4}$$

$$f(x,y) \cdot h(x,y) \leftrightarrow DFT \leftrightarrow F(u,v) \star H(u,v) \tag{2-5}$$

Is important to note, both equations (2-4) and (2-5) also present the same relation if we consider the iDFT transformation, it means that the convolution $F(u,v) \star H(u,v)$ has its representation in spatial domain as the product $f(x,y) \cdot h(x,y)$, which represents the iDFT for $F(u,v)$ and $H(u,v)$.

Now, it possible to apply the Laplacian definition in spatial domain and obtain its representation in frequency domain. The Laplacian kernel and its implementation is show in the equation (2-6) and (2-7), the Laplacian image $g(x,y)$ is shown in the Figure 2-3, which is the Laplacian image using a kernel approach in spatial domain.

$$g(x, y) = f(x, y) \star k_{laplacian}(x, y) \tag{2-6}$$

$$k_{laplacian}(x, y) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2-7}$$

In other hand, once it has been defined the function $g(x, y)$ is possible to obtain the function $G(u, v)$ using DFT. Thus the function $G(u, v)$ corresponds to the Laplacian image in frequency domain which is shown in the Figure 2-4. It is important to note that these Laplacian images, in spatial and frequency domain are based on a kernel or mask approach.

Figure 2-3 Laplacian by Kernel approach in spatial domain for strawberry grey blurred Image

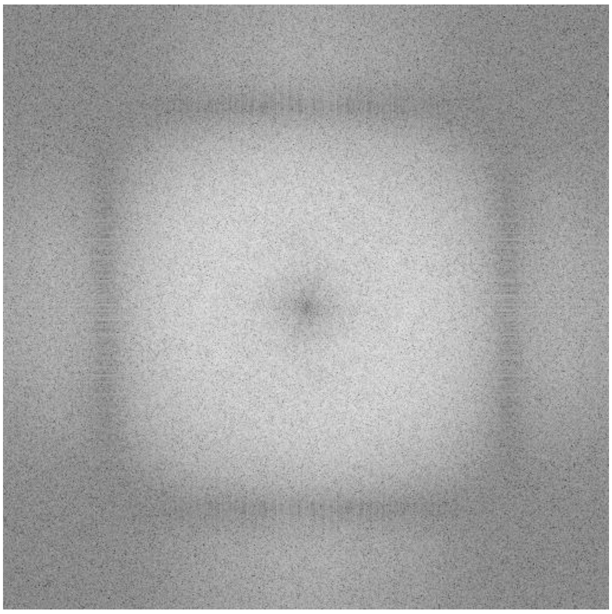


Figure 2-4 Laplacian by Kernel approach in frequency domain for strawberry grey blurred image

The assigned task suggests that the Laplacian filter can be implemented by the function $H(u, v)$, which is described by the equation (2-8).

$$H(u, v) = k(u^2 + v^2) \tag{2-8}$$

In order to build the $H(u, v)$ let's consider the following premises:

1. The implementation of $H(u, v)$ as a filter for the image $F(u, v)$ is described by the equation (2-9).

$$G(u, v) = F(u, v) \cdot H(u, v) \tag{2-9}$$

2. The size for the image $H(u, v)$ must be the same that $F(u, v)$
3. The origin of the frequency reference $\{u, v\}$ is defined in the center of the image $H(u, v)$

It can also be shown that

$$DFT\left\{\frac{d^n}{dx^n}f(x)\right\} = (ju)^n F(u) \tag{2-10}$$

$$DFT\left\{\frac{d^2 f(x,y)}{dx^2} + \frac{d^2 f(x,y)}{dy^2}\right\} = (ju)^2 F(u,v) + (jv)^2 F(u,v) \tag{2-11}$$

$$DFT\{\nabla^2 f(x,y)\} = -(u^2 + v^2)F(u,v) \tag{2-12}$$

Then, the function $H(u,v)$ can be defined as $-(u^2 + v^2)$ for Laplacian filter implementation in frequency domain. Thus, the first step is to build the $H(u,v)$ function following the equation (2-13) and the premise described before. The Figure 2-5 shows the image $H(u,v)$ in frequency domain.

$$H(u,v) = -(u^2 + v^2) \tag{2-13}$$

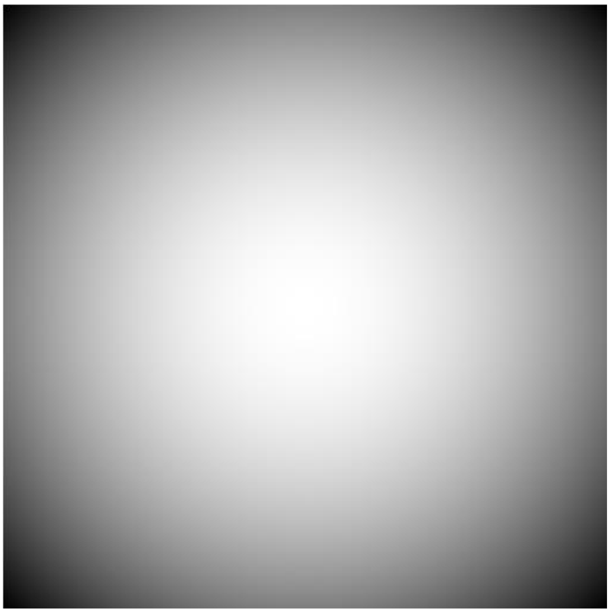


Figure 2-5 Laplacian Image $H(u,v)$ in frequency domain

The equation (2-9) shows the implementation for the $H(u,v)$ filter, for that it is necessary to get the function $F(u,v)$ applying DFT to the image $f(x,y)$ and then calculate the product $H(u,v)F(u,v)$. The Figure 2-6 and Figure 2-7 show the functions $F(u,v)$ and $H(u,v)F(u,v)$ respectively.

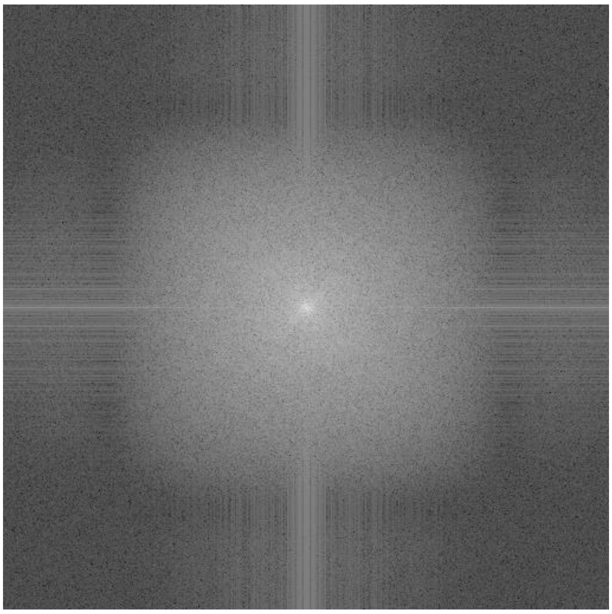


Figure 2-6 Function F(u, v) for strawberry grey blurred Image in frequency domain representation

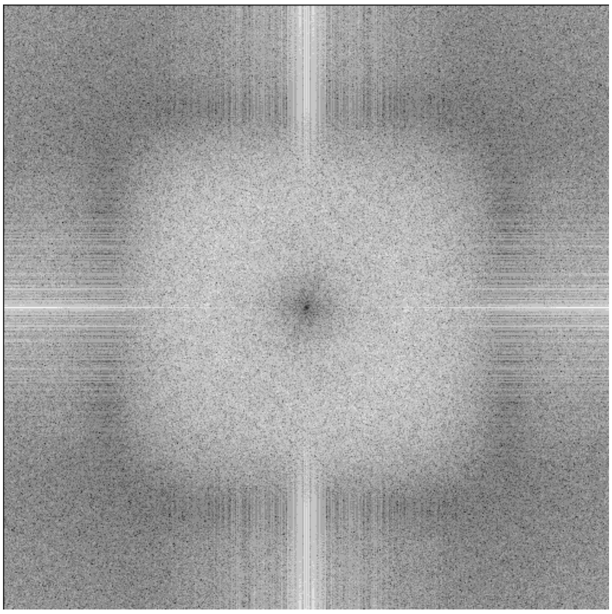


Figure 2-7 Laplacian filter $H(u, v)$ for strawberry grey blurred Image in frequency domain.

The Figure 2-7 in itself is the implementation of the Laplacian filter using the function $H(u, v)$. That implementation defines the function $G(u, v)$, as can be seen in the equation (2-9). This function is the same Laplacian function which was shown in the Figure 2-4 but different approach.

Finally, applying iDFT to the function $G(u, v)$ allows us to get its representation $g(x, y)$ in spatial domain, which is excepted to be similar to the reached one at Figure 2-3 for kernel approach implementation. This Laplacian images is shown in the

Figure 2-8 Laplacian image for strawberry grey blurred Image using $H(u, v)$ representation

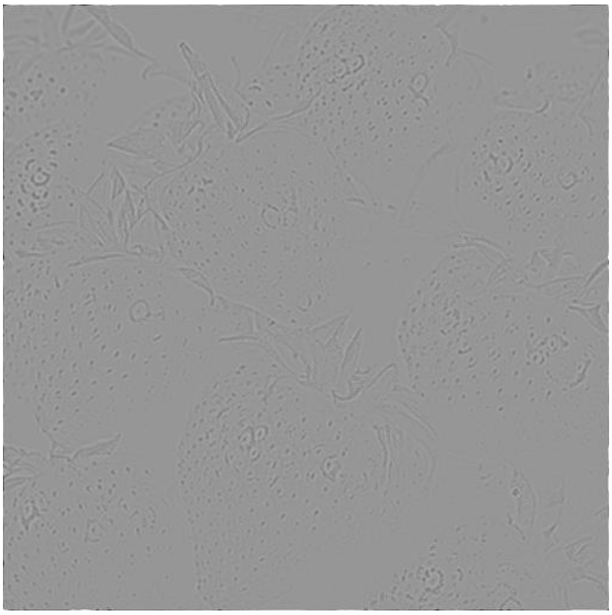


Figure 2-9 Laplacian image scaling for strawberry grey blurred Image using $H(u, v)$ representation

Algorithm

The following algorithm shows how was implemented the function $H(u,v) = -(u^2 + v^2)$ which is the Laplacian representation in frequency domain.

Function: **Laplacian function** (***parameter:*** size)
{
    ‘ the number of rows and cols are defined based on the input parameter size
        rows, cols = ***getting image size(size)***

    ‘Define an output image based on the size passed as parameter
        image = *matrix of zeros*[rows, cols]

    ‘Define the center of the image since definition of the frequency domain
        refer_u, refer_v = rows / 2, cols / 2

    ‘Fill the image output by two nested for loops
    for u in [0, 1, 2 …to refer_u):
        for v in [0, 1, 2 …to refer_v):
            value = -(u*u+v*v)
            image[refer_u - u, refer_v - v] = value
            image[refer_u + u, refer_v - v] = value
            image[refer_u - u, refer_v + v] = value
            image[refer_u + u, refer_v + v] = value

    ‘Scaling the returning image
} **return** image/{.min()-.max()}

Results.

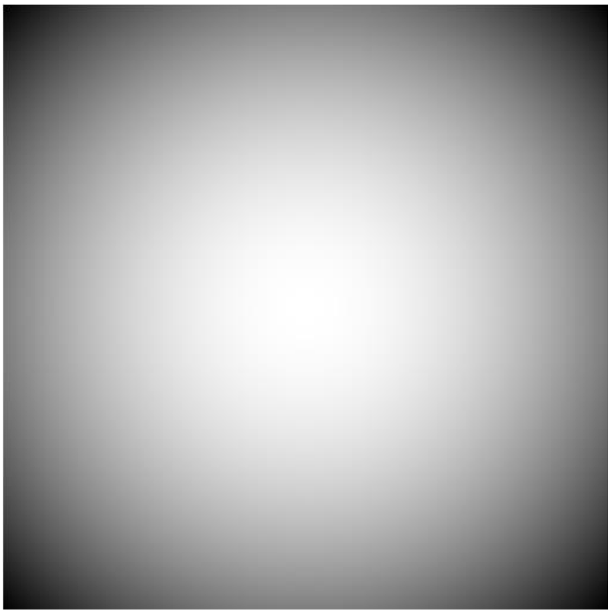


Figure 2-10 Frequency response for function $H(u,v)$

Figure 2-11 Laplacian image for strawberry grey blurred Image by kernel approach

Figure 2-12 Laplacian image for strawberry grey blurred Image using $H(u, v)$ representation

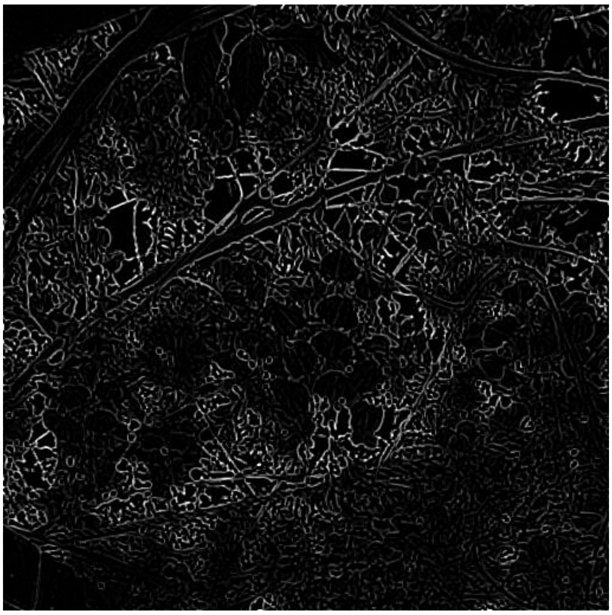


Figure 2-13 Laplacian image for wisteria vine grey blurred Image by kernel approach

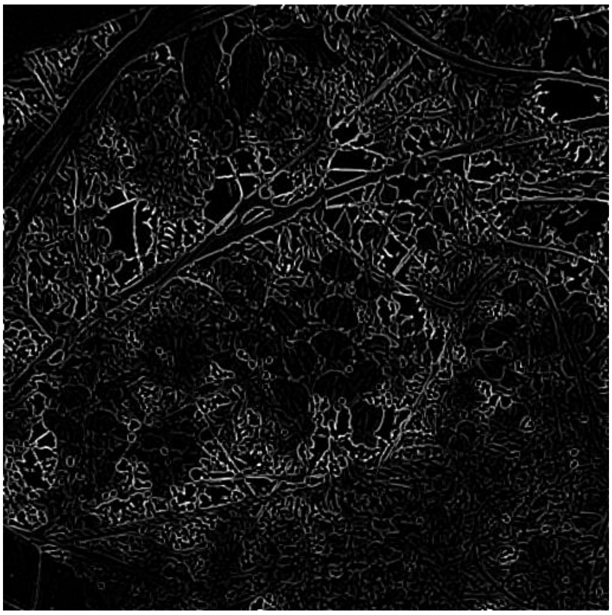


Figure 2-14 Laplacian image for wisteria vine grey blurred mage using $H(u, v)$ representation

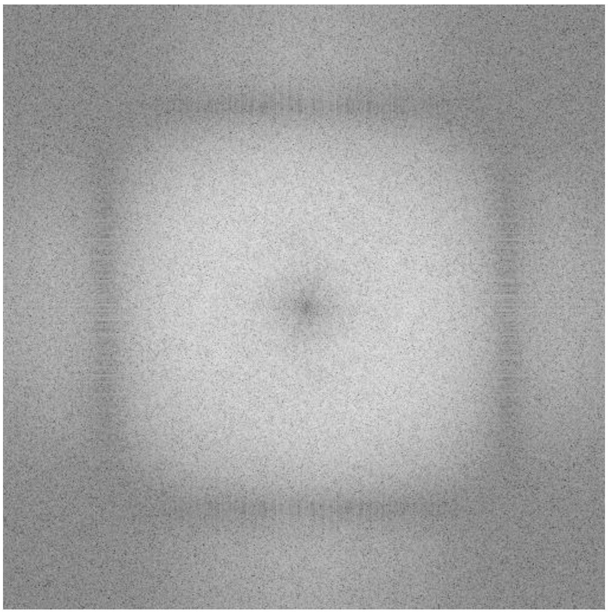


Figure 2-15 Laplacian image for strawberry grey blurred Image by kernel approach in frequency domain

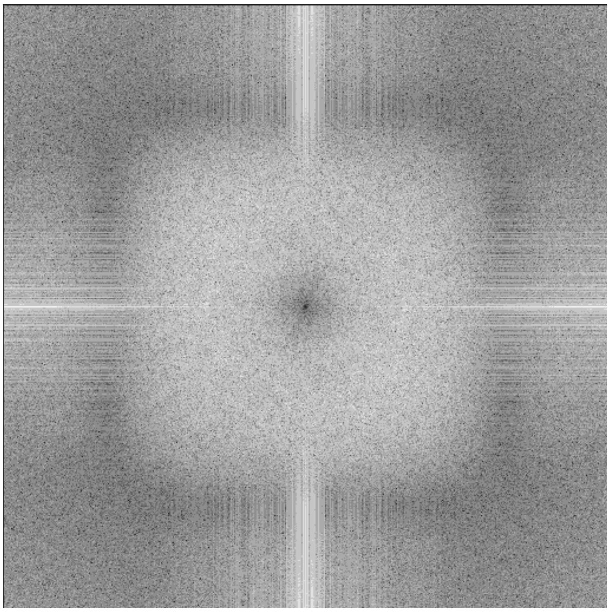


Figure 2-16 Laplacian image for strawberry grey blurred Image using $H(u,v)$ representation in frequency domain

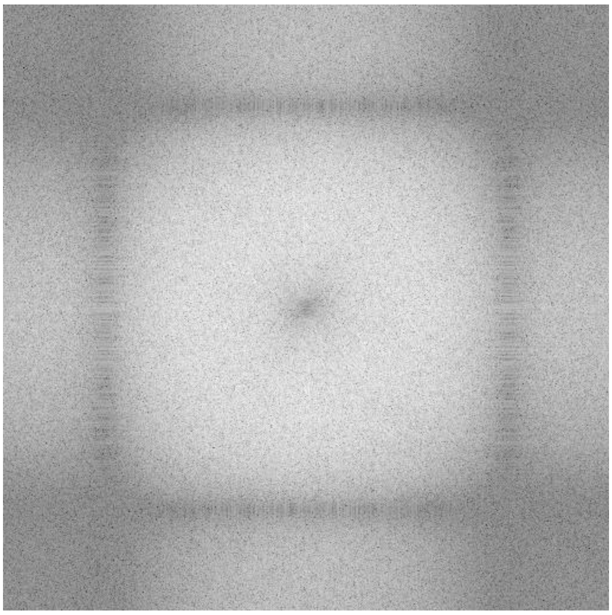


Figure 2-17 Laplacian image for strawberry grey blurred Image by kernel approach in frequency domain

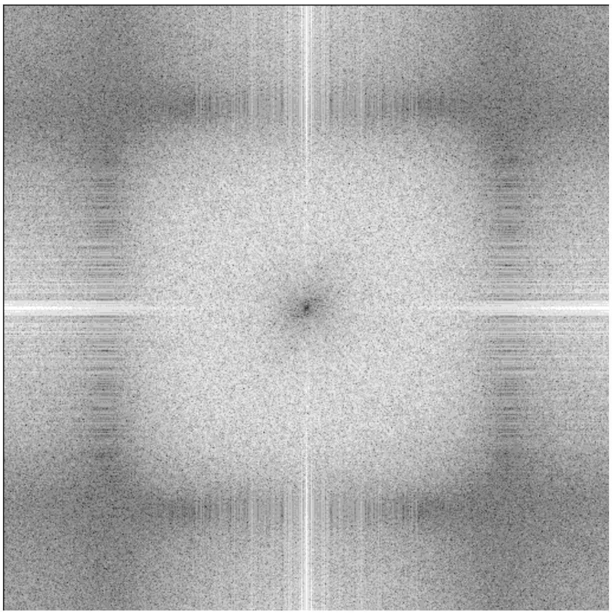


Figure 2-18 Laplacian for strawberry grey blurred Image using $H(u,v)$ representation in frequency domain

Discussion

This section aims to compare the results for two differences way of Laplacian implementation. One by spatial domain using a kernel matrix and convolution operation; and the second one based on frequency domain. For both implementation is possible to appreciate the same response and same results. Thus, is possible to confirm that both implementations are equivalents

3. Section III

Assigned task: For the blurred images **strawberry grey blurred 1** and **wisteria vine grey blurred 1**, try to restore the original image by inverse filtering (deconvolution) method provided that the image was blurred by Gaussian model.

Figure 3-1 Strawberry grey blurred 1 image

Figure 3-2 Wisteria vine grey blurred 1 image

Background

This section focus on an image restoration problem for two given images (Figure 3-1 and Figure 3-2). Before addressing the solution let's define some essential concepts related to this kind of problem.

The model behind of the image degradation can be seen as a function degradation $h(x,y)$ over the original image $f(x,y)$ plus a noise $n(x,y)$. This concept is described graphically in the Figure 3-3. The given images described in the Figure 3-2 and Figure 3-3 correspond to degraded images $g(x,y)$ for the model of degradation. The objective for this kind of problems is to obtain the image $f(x,y)$ or an approximation of it .

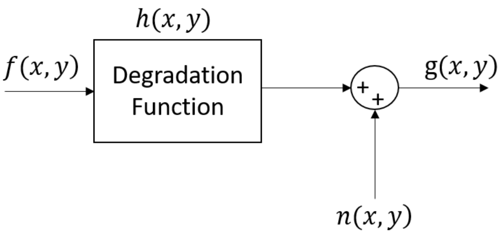


Figure 3-3 Model of image degradation

For this approach a noise-free problem is considered. That means the function $f(x,y)$ has been only distorted by the function $h(x,y)$. Another important assumption is the nature of the function $h(x,y)$, which is defined as a Gaussian model for this problem.

These assumptions allow us to define the following equations

$$f(x,y) \star h(x,y) = g(x,y) \tag{3-1}$$

$$F(u,v) \cdot H(u,v) = G(u,v) \tag{3-2}$$

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

$$H(u,v) = e^{-\frac{1}{2}\left[\frac{D(u,v)}{\sigma}\right]^2} \tag{3-4}$$

$$D(u,v) = \sqrt{u^2 + v^2} \tag{3-5}$$

The equation (3-1) defines the degraded image $g(x,y)$ as the convolution between the degradation function $h(x,y)$ and the original image $f(x,y)$, which in itself is the target for this section. It is also possible to define $G(u,v)$ as the product of the function $F(u,v)$ and $H(u,v)$ in frequency domain. Thus, the function $F(u,v)$ can be defined as the product of the function $G(u,v)$ and the $H(u,v)^{-1}$. Furthermore the definition of a Gaussian function in frequency domain is shown by the equation (3-4), where σ is the standard deviation for the Gaussian model and $D(u,v)$ is the Euclidean distance defined by the equation (3-5).

In order to get the function $F(u,v)$ the equation (3-3) suggest inverting the Gaussian model. However, since some values of the Gaussian model can be close to zero, inverting these elements would give us either infinities or extremely high values for the function $F(u,v)$. Thus, define a threshold for such values can avoid these problem. The implementation of the inverted Gaussian model is defined as following.

$$H(u,v)^{-1} = \begin{cases} \frac{1}{H(u,v)} & if \ \frac{1}{H(u,v)} < \varepsilon \\ \varepsilon & else \end{cases} \tag{3-6}$$

Algorithm

The following algorithm shows how was implemented the function $H(u,v) = -\frac{1}{2}\left[\frac{D(u,v)}{\sigma}\right]^2$ which is the Gaussian representation in frequency domain.

Function: **Gaussian function** (***parameter:*** size, epsilon, sigma)

{

' the number of rows and cols are defined based on the input parameter size

rows, cols = ***getting image size(size)***

'Define an output image based on the size passed as parameter

image = *matrix of zeros*[rows, cols]

'Define the center of the image since definition of the frequency domain

```
        refer_u, refer_v = rows / 2, cols / 2
```

'Fill the image output by two nested for loops

```
for u in [0, 1, 2 …to refer_u):
        for v in [0, 1, 2 …to refer_v):
                ' Ecuclidean distance
                d = dist_(u, v)
                fq = exp(-d/(2*sigma^2))
                image[crow - u, ccol - v] = fq
                image[crow + u, ccol - v] = fq
                image[crow - u, ccol + v] = fq
                image[crow + u, ccol + v] = fq

image = image/{.min()-.max()}
```

'***For all values less than epsilon overwrite the value of epsilon***

```
image[image <= epsilon] = epsilon
```

'***Return image in frequency domain***

}***return*** image

Results

Figure 3-4 Degraded Image for Strawberry grey blurred 1 image

Figure 3-5 Restored Image for Strawberry grey blurred 1 image

Figure 3-6 Degraded Image for Wisteria vine grey blurred 1 image

Figure 3-7 Restored Image for Wisteria vine grey blurred 1 image

Discussions

For this section a restoration problem was proposed. Furthermore, a Gaussian model is assumed as function $H(u,v)$, which is called degradation function. The used solution for this problem was the Inverse Filtering.

The implementation of this algorithm define some parameters which define the response and quality for the restored image. These parameters are listed as following:

1. Epsilon ε, which defines a threshold to avoid values close to zero.
2. Sigma σ, which define the standard deviation for the Gaussian model.

A proper tradeoff between these parameters allows to reach a good restored image. By contrast values too low or high produce poor results.

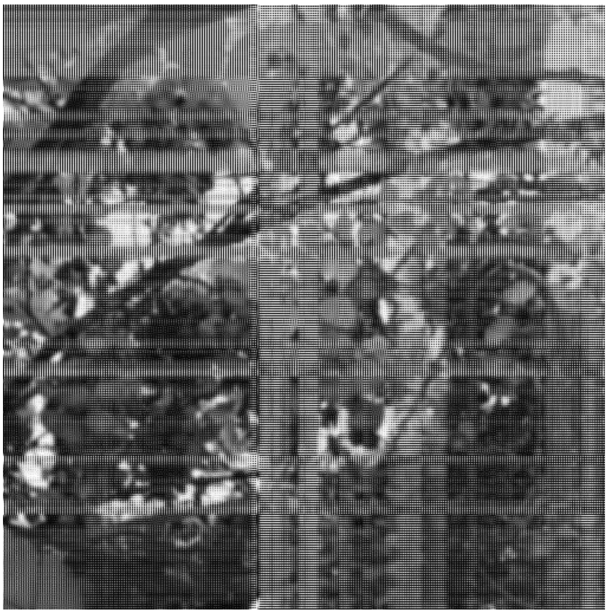


Figure 3-8 Restored image with Epsilon ε parameter too low

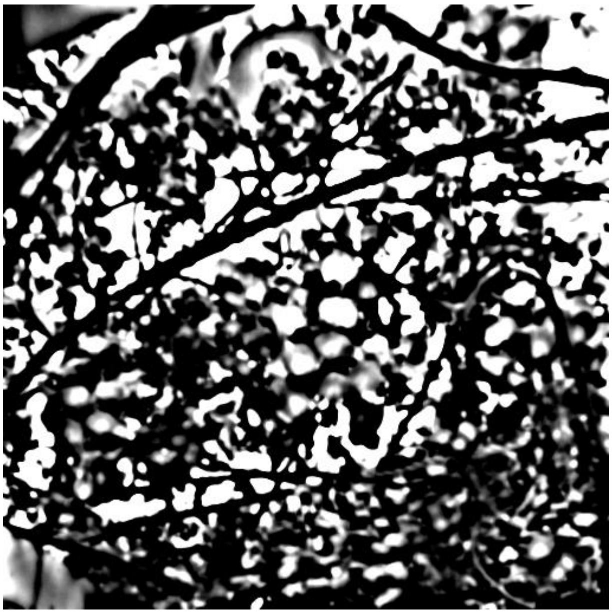


Figure 3-9 10 Restored image with Sigma σ parameter too low