Laboratory Report II
Image filtering and Fourier Transform

DIBRIS, Robotics Engineering, 2021
University of Genoa

Bauyrzhan Zhakanov
bauyrzhan.zhakanov@gmail.com

27 October 2021

Abstract

The laboratory work is dedicated to study and experiment with the image filtering techniques. In this laboratory, we are going to study how to add noise using the Gaussian standard deviation and noise density, although its removal using Gaussian and median filters and compare the pixels difference on 3x3 and 7x7 masks. Moreover, we are going to work on some linear filters from 41-45 pages of the presentation, and apply Fourier Transforms to study the transformed images with different parameters such as 101x101 pixels with sigma=5, 101x101 pixels with sigma=5. The results of every experiment is represented during the slides.

1 Introduction

An image is comprised of individual pixels, can be considered as a function, where each pixel also has its own value. In terms of a grayscale image, each pixel has its own an intensity between 0 and 255, where 0 is being black and 255 is being white. Position of the pixels is represented with (x, y) coordinates, and f(x,y) give the intensity of the image at pixel position.

In this laboratory, the image filtering techniques are studied: - adding noise with Gaussian standard deviation of 20% and salt&pepper with 20% density - its removal using 3x3 and 7x7 masks by moving average, Gaussian and median filters - using the convolutions filter with 7x7 matrix from 41-45 pages - representing Fourier Transforms with 101x101 spatial support with sigma 5 for transformed images

Every filtered image is depicted with a help of imagesc() and surf() function on MATLAB.

2 Noise addition

In this laboratory exercise, the noise is added to the original image. It is also known as impulse noise, considered as black and white dots on the image.

As it can be seen from the pictures of Figure 1, images with 20% of standard deviation of noise can be seen clearly than salt&pepper, since the black and white dots are quite larger in the below one. Although, the histograms of images tells that the intensity values near 0 and 255 have almost 0 values, while the salt&pepper histogram has large the intensity values from both 0 and 255 sides. That is why it is possible to observe white and black dots.

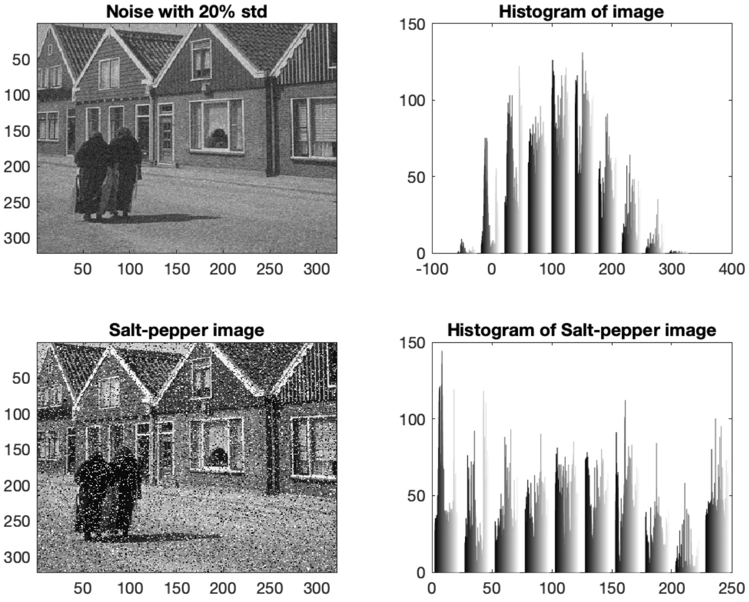

Figure 1: Comparing noises from standard deviation and salt&pepper

3 Noise removal

There are 3 methods for removing the noise. As an example, the salt&pepper image is used to demonstrate their differences: moving average, low-pass gaussian filters, and median filters. Although, 3x3 and 7x7 masks are used to show how filters might be used to remove noise.

3.1 Moving average filter

As it can be observed from the pictures of Figure 2, using 7x7 masks has more blurring than 3x3. The histograms of 7x7 removes the values near to 0 and 255, so the surf() function shows more smoothed picture; while the 3x3 has more values near to 0 and 255, so the surf() function shows more sharpened 3-D picture.

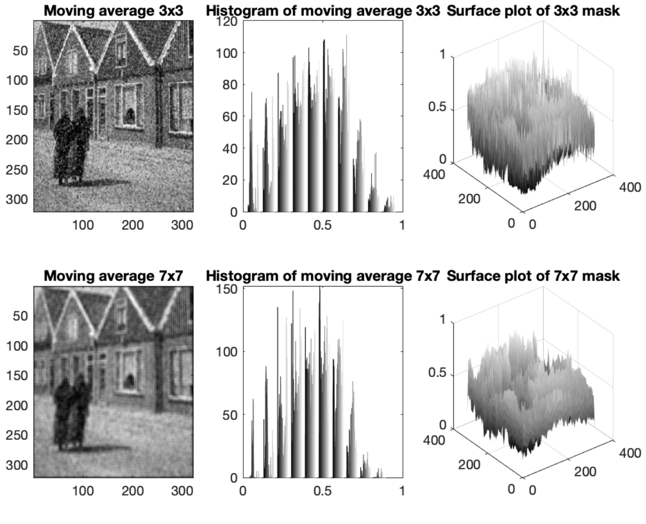

Figure 2: Moving average filter to remove noise

3.2 Low-pass gaussian average filter

As the next method, low-pass Gaussian filter is used to remove the noise. The results of gaussian filters has shown in Figure 3, and has more sharpened values on surf() function. Observing the histogram, the values near to 0 and 255 is not removed, while the magnitude of the 7x7 mask has less value than 3x3.

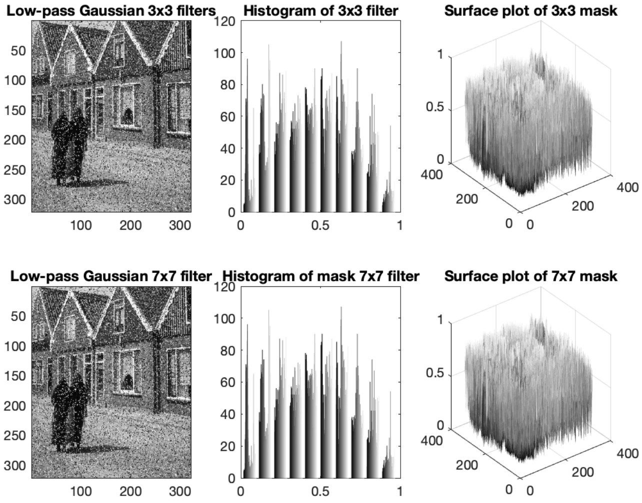

Figure 3: Low-pass Gaussian filter to remove noise

3.3 Median filter

The median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring pixels. As observed from Figure 4, results looks better than previous both filters.

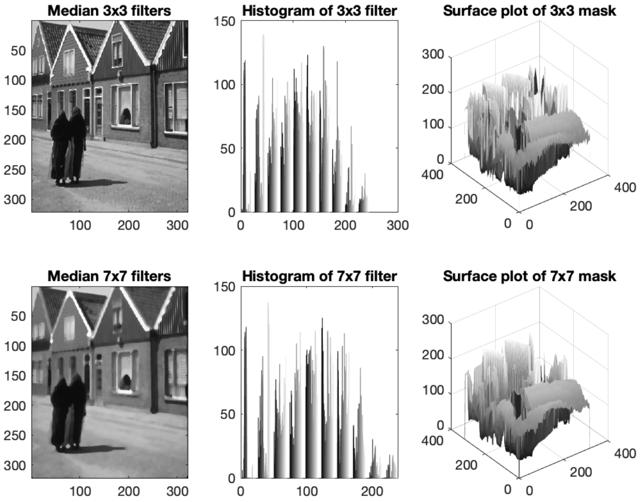

Figure 4: Median filter to remove noise

For small to moderate noises, the median filter depicts better results than the low-pass gaussian one at removing noisy dots by saving the edges for a given image. The histograms has less values near to 0 and 255 in comparison with low-pass gaussian filter, but more than moving average filter. Although, the surf() function is much smoother than the previous ones.

4 Linear filters

In these experiments, the several linear filters are applied to observe the image transformations based on the 7x7 matrices. Those implementations are taken from the pages from 41 to 45 of the presentation: filtered image with no change, shifted image, blurred image, and a sharpened image. In the page 45, the sharpening method is used by smoothing and detailing the original image.

4.1 Filtered with no change

In this experiment, the image is added by 7x7 matrix with 1 in the center and zeros around. It make no changes since the matrix can not change the intensity values. The results has shown in Figure 5.

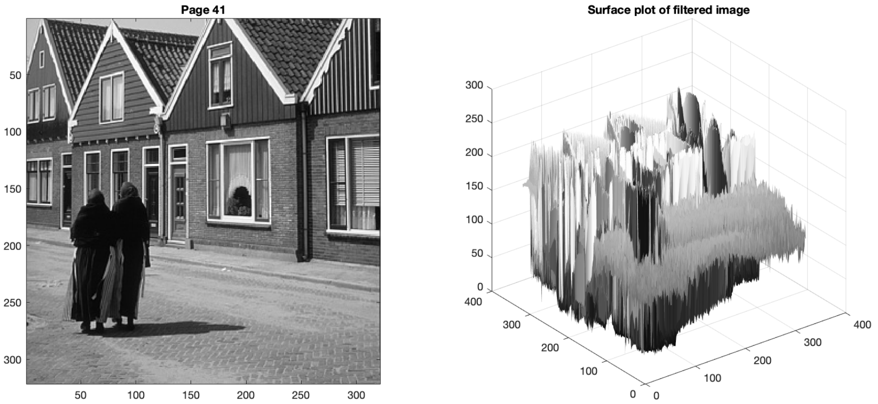

Figure 5: Filtering from page 41

4.2 Filtered and shifted

In this experiment, the image is added by 7x7 matrix with 1 in the most right and zeros around. As it can be seen from the surface plot, the right side is black colored, because the intensity values are multiplied to 0. The results has shown in Figure 6.

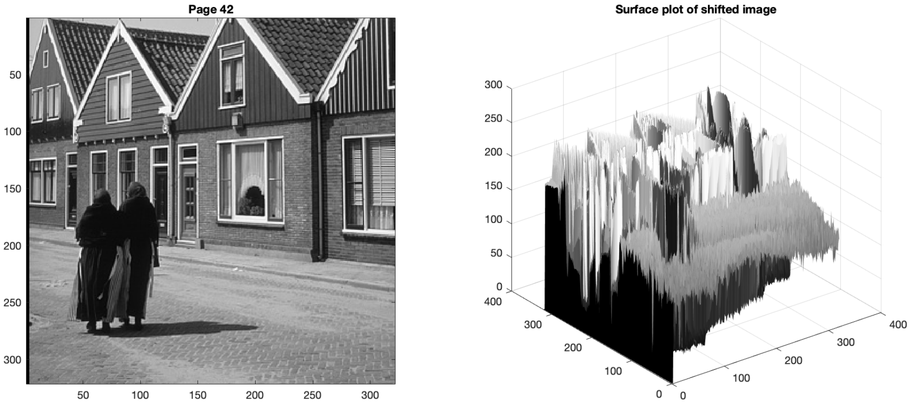

Figure 6: Filtering from page 42

4.3 Blurred with a box filter

In this experiment, the image is added by 7x7 matrix with ones. As it can be seen from the surface plot, the image is blurred. The data points of an image are modified so each points higher than the adjacent pixel with sharped points are reduced. The results has shown in Figure 7.

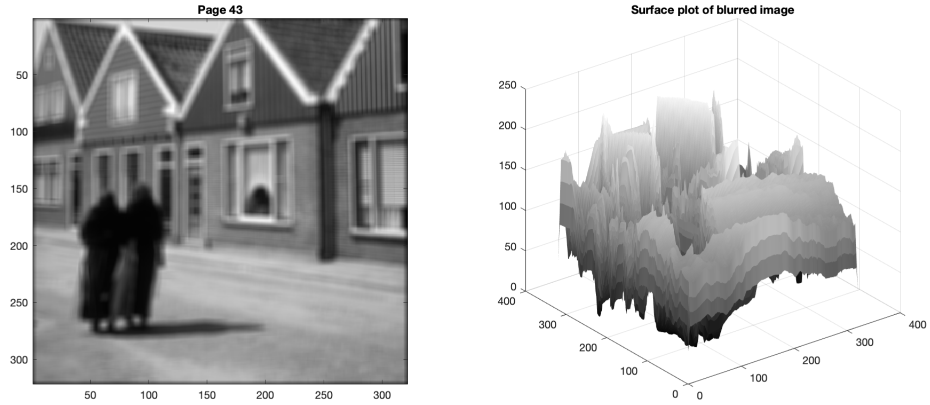

Figure 7: Filtering from page 43

4.4 Sharpening filter

In this experiment, the image is added by 7x7 matrix with ones. As it can be seen from the surface plot, the image is highlighting the edges of the highest intensity values. The results has shown in Figure 8. By looking at the code:

```
% Page 44
filter_44_m1 = filter_41 * 2;
filter_44_m2 = ones(7)/49;
filter_44 = abs(filter_44_m1 - filter_44_m2);
filtered_44 = conv2(img,filter_44, 'same');
```

In the code above, the first line *filter_41* is demonstrating the 1 at the center and zeros around 7x7 matrix.

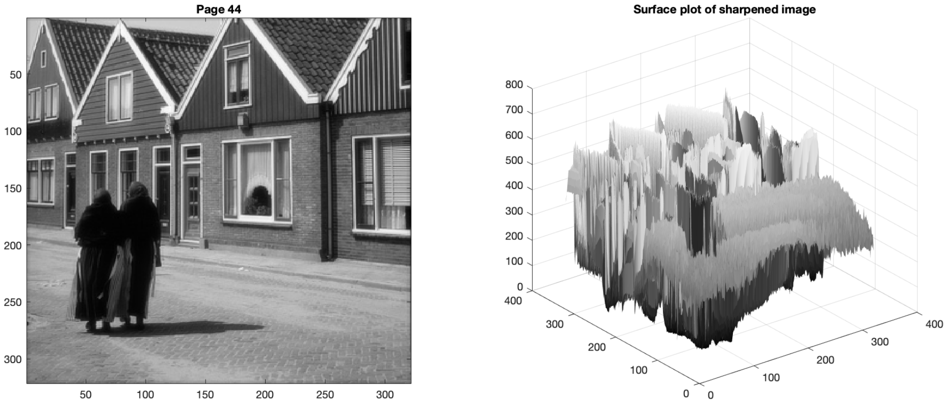

Figure 8: Filtering from page 44

4.5 Sharpening filter from page 45

The another way of the sharpening image is to use the smoothed and detailing the image. As it can be seen from the surface plot, the image is detailed and smoothed. As the sharpening value 1/10 is given. The results has shown in Figure 9. By looking at the code:

```
% Page 45
K = ones(7)/49;
smooth = conv2(img,K, 'same');
detail = img - smooth;
a = 1/10;
sharp = img + a*detail;
```

As it can be seen from the Figure 9, the image is smoothed. After the detailed image is taken from the difference of original image from smoothed image. Addition of both original images and detailed image with a value 1/10 is giving a sharpened image. Based on the surface plot function, the neighbors have strict changes.

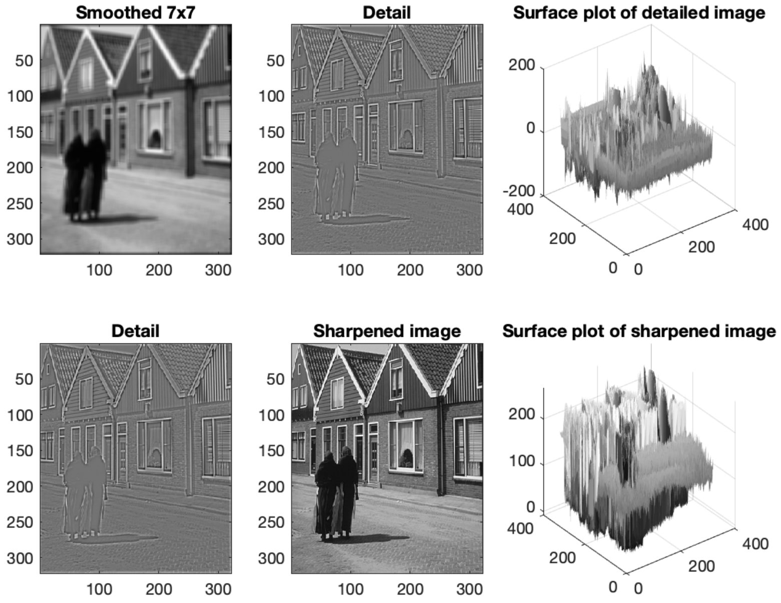

Figure 9: Sharpening steps from page 45

5 Fourier Transforms

As it can be illustrated from the transformed images, the output from the Figure 10 represents the image in the frequency domain. The filtered and shifted images has no differences, while the Gaussian blurred image has concentrated in the center frequencies in the spatial domain image. This means that their magnitude gets smaller for higher frequencies. The reconstruction into the sharpening has a bit more points towards center.

In Figure 11, sing the Gaussian filter with 101x101 pixels with sigma 5 has shown the black colored image with white dot inside. As it can be seen, the intensity values are high of the image.

In Figure 12, the filter with a spatial support of 7x7 pixels, and copied it in the middle of a zeros image of 101x101 pixels has shown quite different results. This means that the intensity values are high around, but lower at the center of the image.

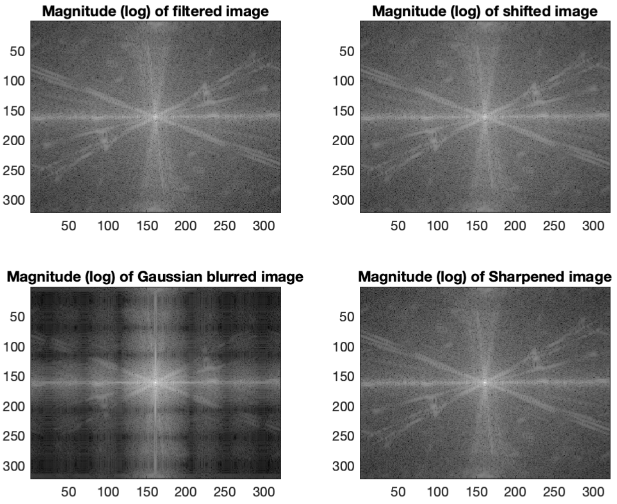

Figure 10: Magnitude of transformed images

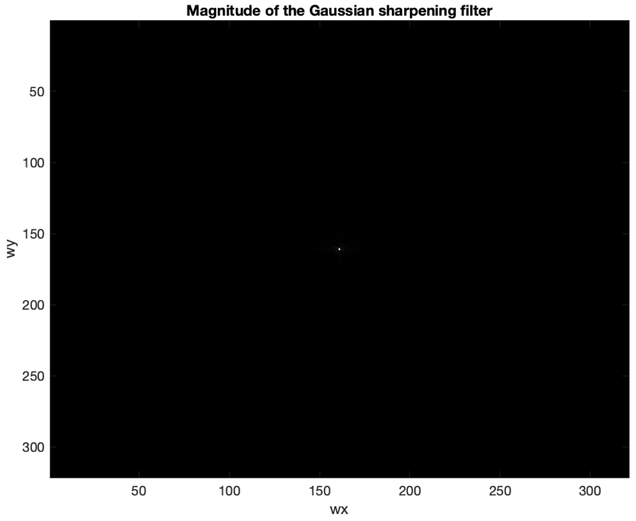

Figure 11: Magnitude of Gaussian filtered image

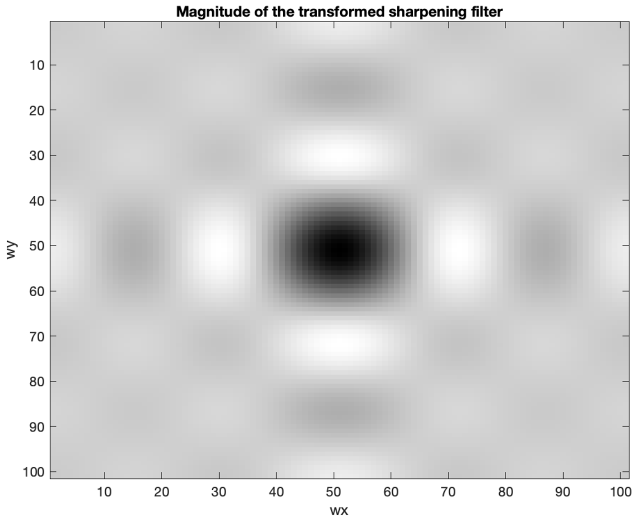

Figure 12: Magnitude of sharpening filter