UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

COMPUTER VISION

First Assignment

Authors:

Marmolino Giorgio s772100
Penna Lorenzo s4953984
Petrosilli Marco s7812048

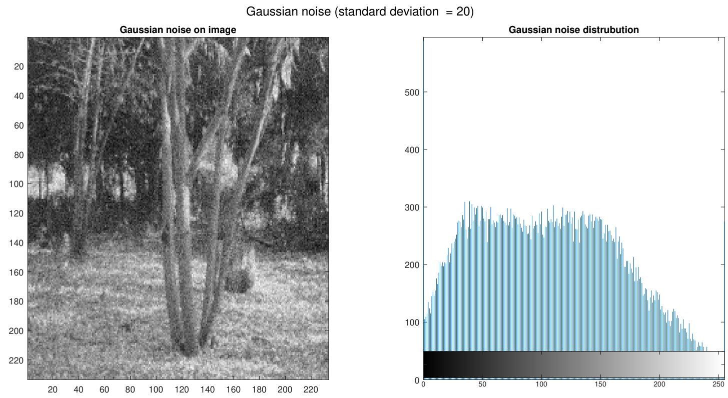
Contents

1 Noise generation

Starting from the input image we will apply two different type of noise:

- gaussian noise (standard deviation 20);
- salt & pepper noise (density 20%);

In this way we will obtain the following results:



Gaussian noise: Pixels in the image are modified based on the Gaussian distribution, meaning that some pixels will become lighter or darker compared to their original value. This causes a slight random "blurring" or "graininess," which alters the sharpness and details of the image. Looking at the histogram of the image, the distribution of the pixels has a Gaussian trend. When the Gaussian noise is intense, fine details in the image (such as edges and textures) may become less defined, giving the image a blurry or unclear appearance; furthermore, after the noise is applied, most of the changes to the pixels will be close to the original value, with few extreme changes. This type of noise mimics real-world disturbances in sensors or image acquisition systems.

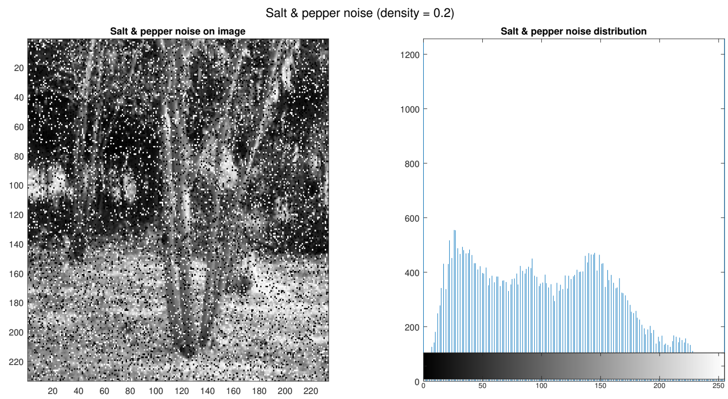


Figure 1: Different type of noise on test image

Salt & Pepper noise: Pixels are randomly replaced with extreme values, i.e., the maximum value for salt (typically 255 in 8-bit images, resulting in a white pixel) and the minimum value for pepper (typically 0 in 8-bit images, resulting in a black pixel); indeed looking at the istogram of the image, now there are two peaks for pixel values of 0 and 255. The noise appears as scattered spots of white and black pixels throughout the image, without a regular pattern. Since Salt and Pepper noise completely replaces the original pixel values with extreme values (0 or 255), the original information in those areas of the image is lost, particularly affecting homogeneous areas and thin edges.
Images corrupted by noise may be more difficult to interpret for artificial vision algorithms, which is why various types of filters can be used to restore the image quality.

2 Task 2 - Moving average, low-pass Gaussian filter and median filter use

After obtaining the images with the noise applied, the low pass Gaussian, moving average and median filter filters are applied in order to try to remove the noise.

2.1 Filters structure

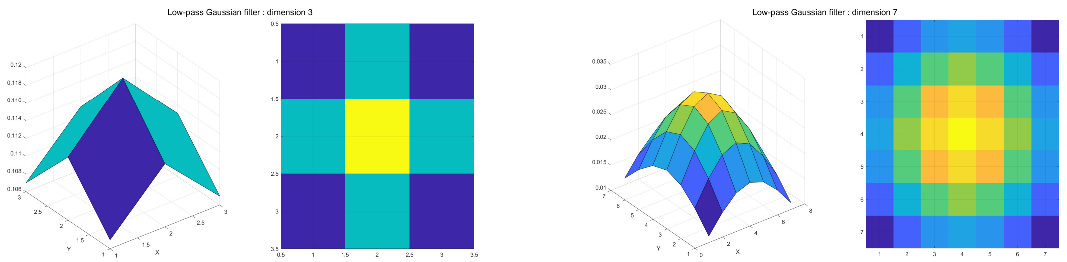
The different types of filters were created by varying the size of the spatial support between 3x3 and 7x7; specifically, the choice of a small spatial support has a localized effect and is more suitable for preserving fine details in the image. On the contrary, a larger spatial support covers a more extensive portion of the image, allowing the filter to detect or influence larger structures in the image (useful for removing noise distributed over a large area or for enhancing large objects).

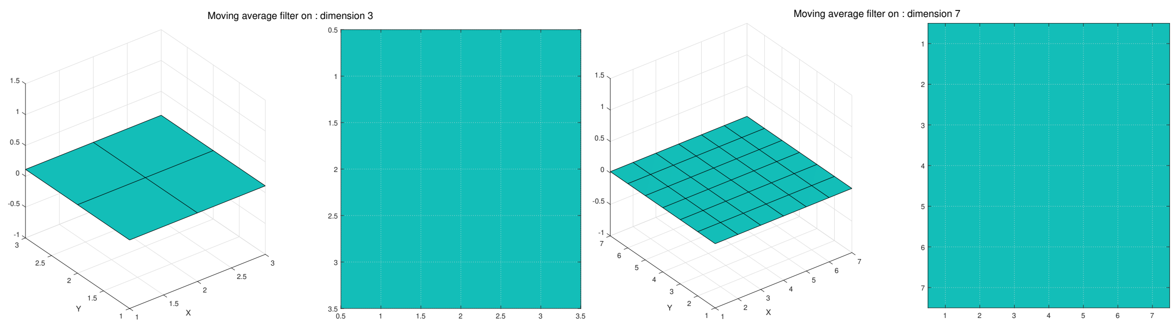
In general, filters with a larger spatial support require more computational operations, as they involve more pixels in the processing, slowing down the application of the filter, especially on large images. Among the filters used in this application there are:

- Gaussian filter: A Gaussian filter with a larger support will have greater "spatial variance," meaning the blurring effect will be more distributed and involve more distant pixels. When the standard deviation is low, the Gaussian filter has a mild blurring effect. The pixels within the Gaussian window that are closer to the central pixel contribute more to the result, while the more distant pixels have less influence. As the standard deviation increases, the Gaussian distribution becomes wider, meaning a larger area of surrounding pixels is taken into account. Pixels farther from the center will have a greater influence in the calculation of the weighted average.

- Median filter: With a larger spatial support, the median filter will better remove "salt-and-pepper" noise but may also alter the image's edges.

- Moving average filter: The moving average filter in images is useful for reducing noise and smoothing, but it can lead to a loss of important details. It is particularly suitable when the goal is to improve the overall visual quality of an image or prepare it for further analysis, provided that the trade-offs in sharpness are considered;

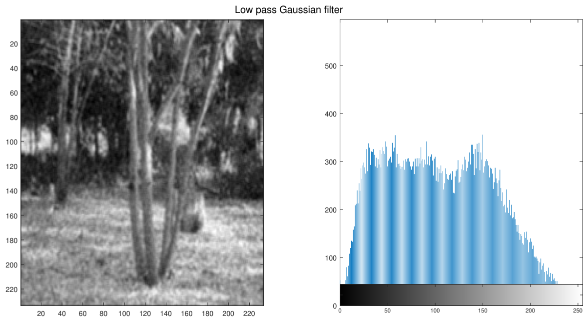


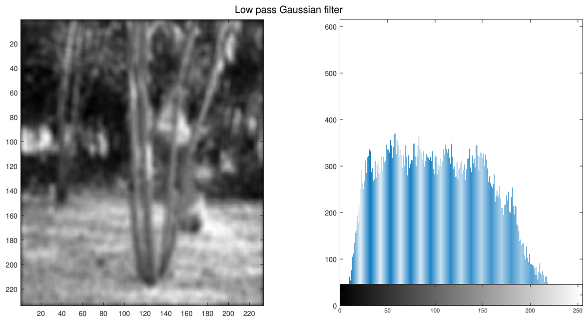


Unlike the others, the median filter is a non-linear filter that does not have a fixed structure, which is why it cannot be plotted.

2.2 Applying filters

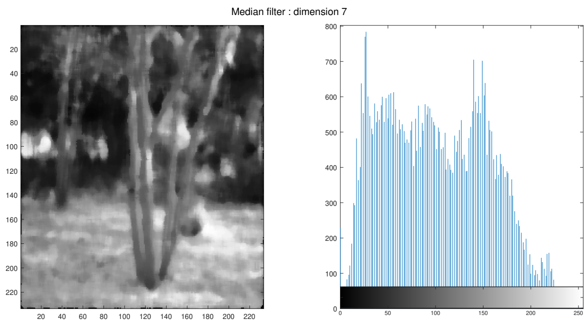
All types of filters are applied on the previous images affected by noise, obtaining the following results; results will be different, depending on the type of filter used, by the dimension of the spatial support and the type of noise affecting the image.

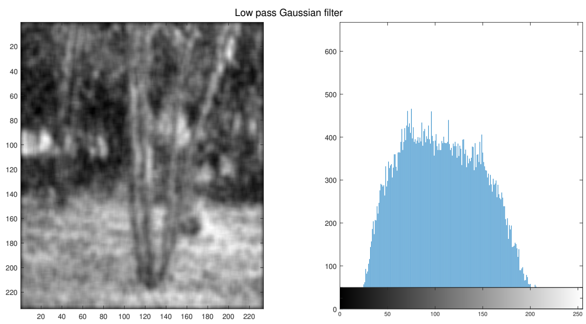




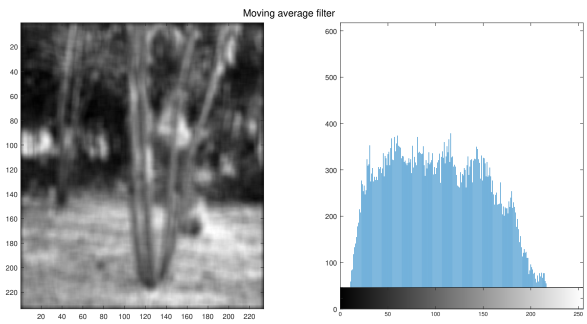
For instance, we can see the difference of using the same type of filter (low pass Gaussian filter) with different spatial support dimension on an image affected by gaussian noise. Using filters with a bigger spatial support we'll lose details on images.

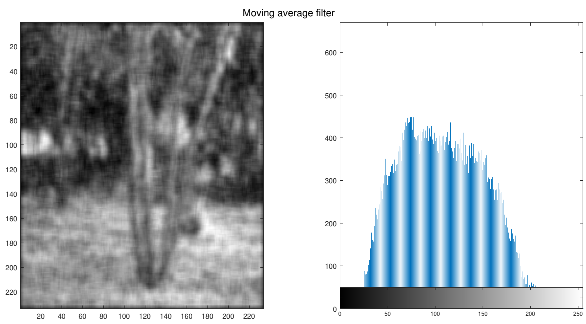
By the way, it is possible to see that based on the type of noise, not all filters are good in the same way. For example, for an image affected by salt & pepper noise, it will be better to use a median filter instead of a gaussian filter:





Here we can see its effect, considering the same spatial support dimension on images affected by different type of noise: as we saw in the previous cases, using a bigger spatial support we'll lose details in the image.





3 Task 2 - Linear filters

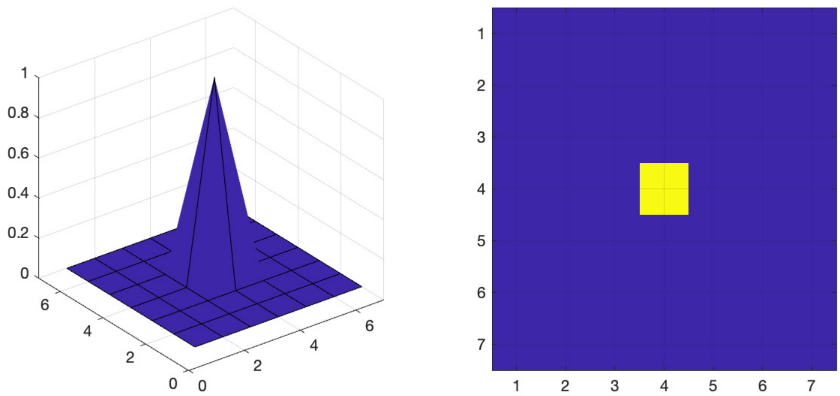
In this part of the lab, starting from an input image we will apply different types of linear convolution filters using a spacial support equal to 7x7, and we will observe how the result changes based on the different ways in which the kernel is built. The image below was used as input to experiment with linear filters.

Original image

3.1 Impulse filter

This filter is obtained building the kernel as follows, this conformation doesn't cause changes on the given image.

Impulse filter

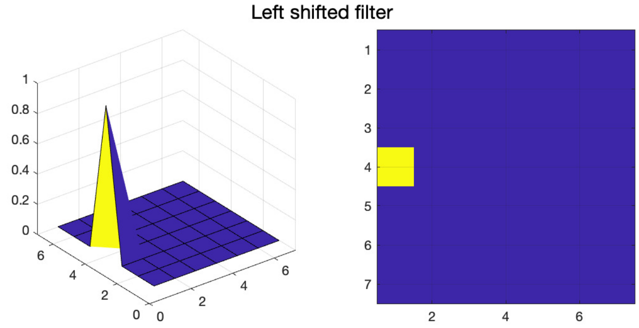


And after the convolution we can appreciate that the result is the same of the given image.

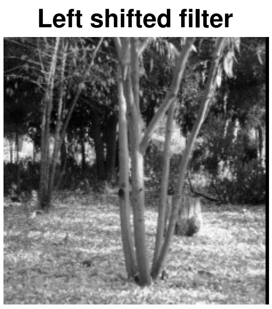
Impulse filter

3.2 Left Shifted filter

This filter, how the name suggests, allow us to shift the given image at the left by 1 pixel, and the "external" column is replaced by a 0-column.

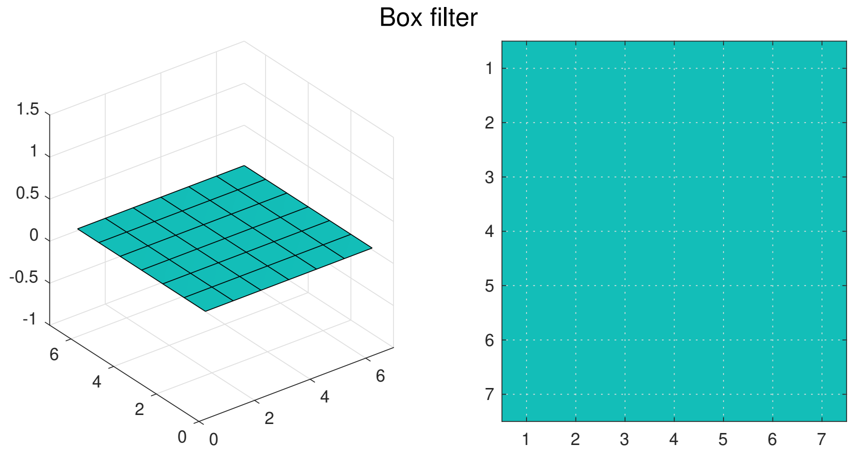


The following result was obtained.

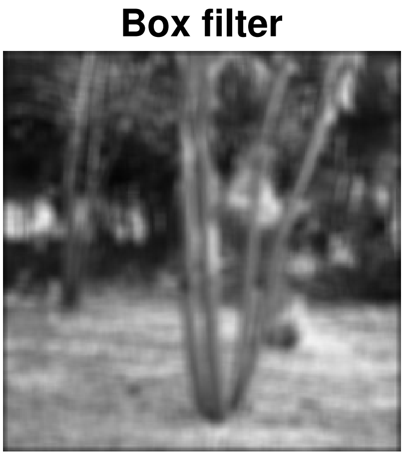


3.3 Box filter

The box filter convoluted on the input image allows us to blur it, all the spacial support units have the same value based on the number of those.

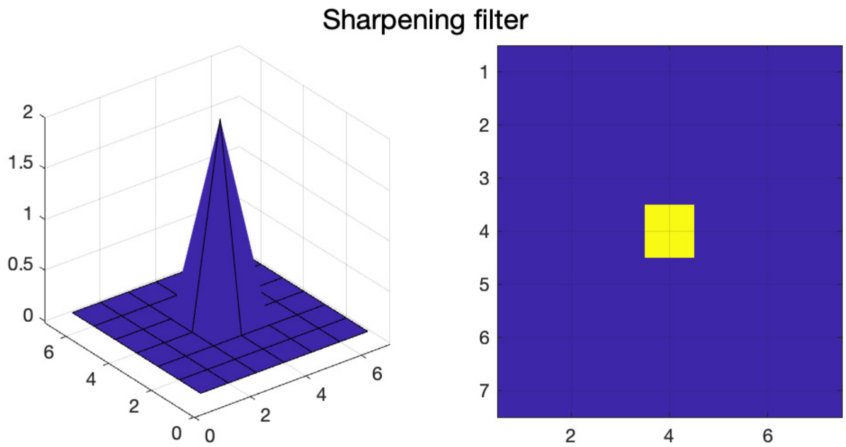


The blurred image is shown below, consider that increasing the spatial support results in more averaging over a larger region of pixels, reducing more noise and fine detail in the image.



3.4 Sharpening filter

This filter's used in image processing to improve sharpness, highlighting the edges and details of images. It works by enhancing rapid variations in intensity, that is, edges between objects or regions with different pixel values. We should build the kernel as shown in the figure.



And after the convolution the sharpened image will appear as follows.

Sharpening filter

3.5 Sharpened image by detailed image

Use a sharpening filter is not the only way to obtain a sharpened image. In fact in this case we subtracted from the real image a blurred image (obtained by box filtering) to obtain image details.
Then we add image details to the real image to increase the intensity of the edges and contrasts in the image, in adding the detailed image is multiplied by a proportionality constant, the bigger this is, the more the contours will stand out.

Detail from image

Sharpened image

4 Task 4 - Fourier transform

The goal of this part of the lab is to apply the Fourier Transform on an image and on various filters . Specifically, we aim to:

1. Display the magnitude spectrum (log scale) of the transformed image.
2. Display the magnitude of a Gaussian low-pass filter (101x101 pixels, sigma = 5) in the frequency domain.
3. Display the magnitude of a sharpening filter (7x7 pixels) placed in a 101x101 zeros matrix, transformed using the FFT.

4.1 Fourier Transform of the Original Image

In the first step, an image (tree.png) was transformed using the Matlab function FFT, so that different frequency components of the image become visible.

The resulting magnitude spectrum was visualized using a logarithmic scale to enhance the visibility of both large and small frequency components. Typically, low-frequency components are concentrated at the center of the spectrum, representing smooth areas of the image, while high-frequency components, responsible for sharp edges and details, are distributed further from the center.

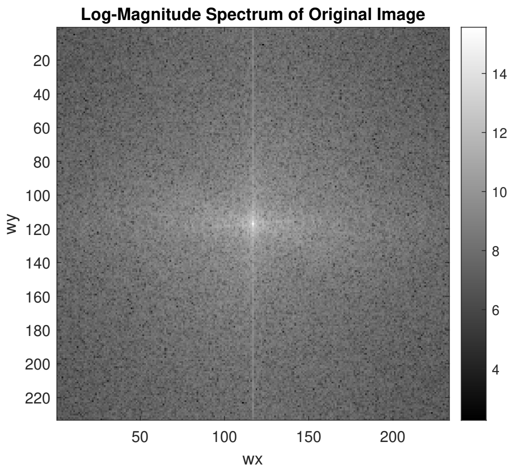


Figure 2: Magnitude of the transformed original image

4.2 Gaussian Low-Pass Filter

The Gaussian low-pass filter was created with a size of 101x101 pixels and a standard deviation (sigma) of 5. The Gaussian filter attenuates high-frequency components, which correspond to rapid changes like noise or edges, allowing only the low-frequency components to pass through.

As we can visualize the Fourier transform exhibits a bell-shaped curve centered at the origin, revealing its low-pass characteristics.

4.3 Sharpening Filter (7x7 on 101x101 Zeros)

A 7x7 sharpening filter was created and placed in the center of a matrix of 101x101 zeros. The sharpening filter is designed to enhance the high-frequency components of an image, which correspond to edges and fine details.

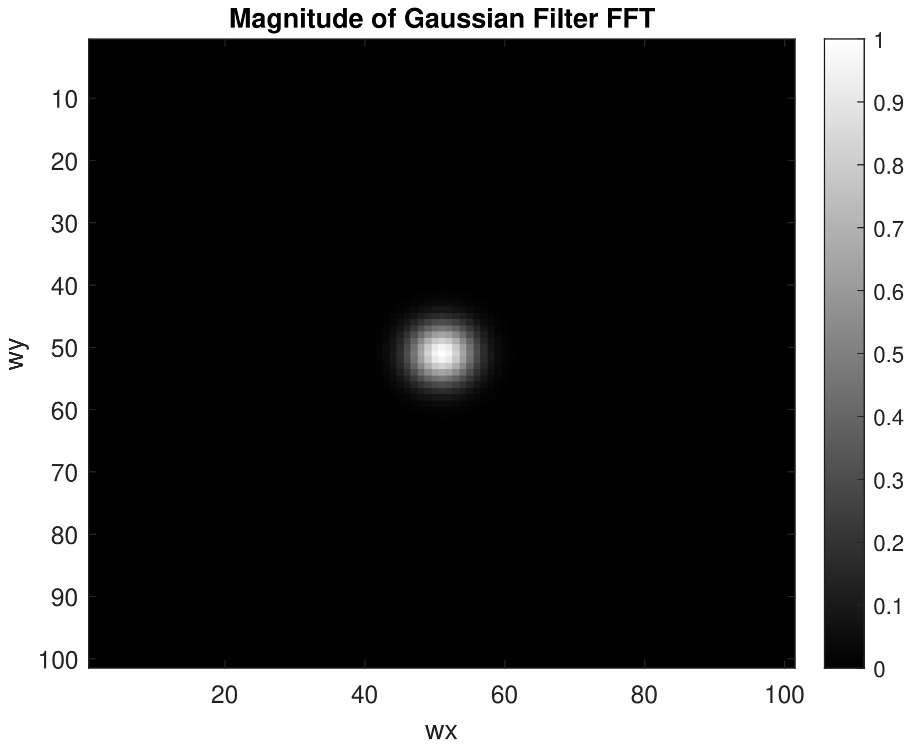


Figure 3: 2D Gaussian Filter

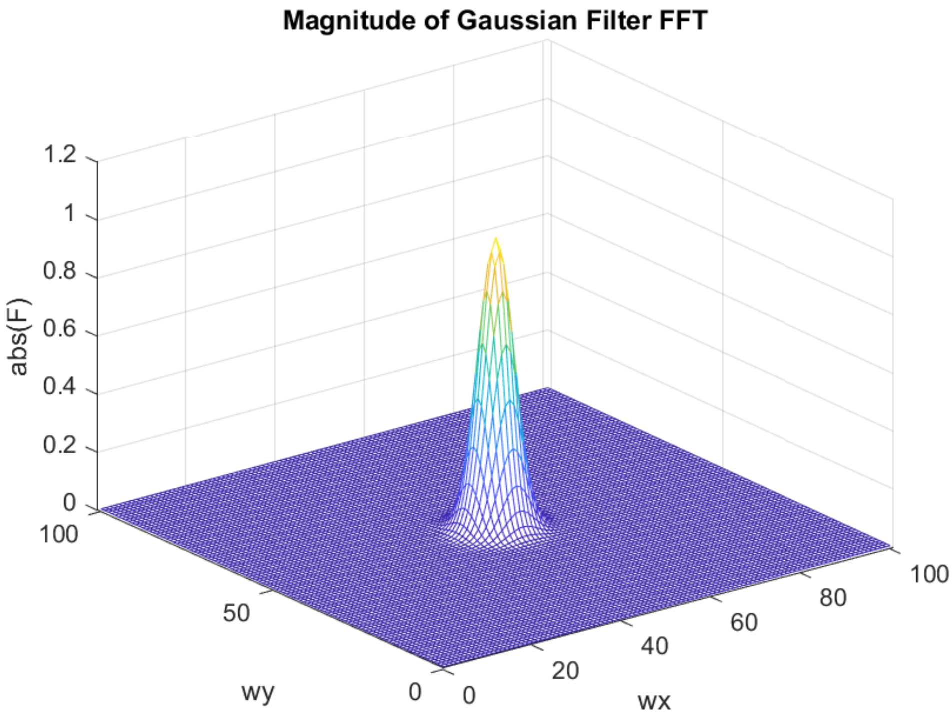


Figure 4: 3D Gaussian Filter

Figure 5: Magnitude of the transformed Gaussian Filter

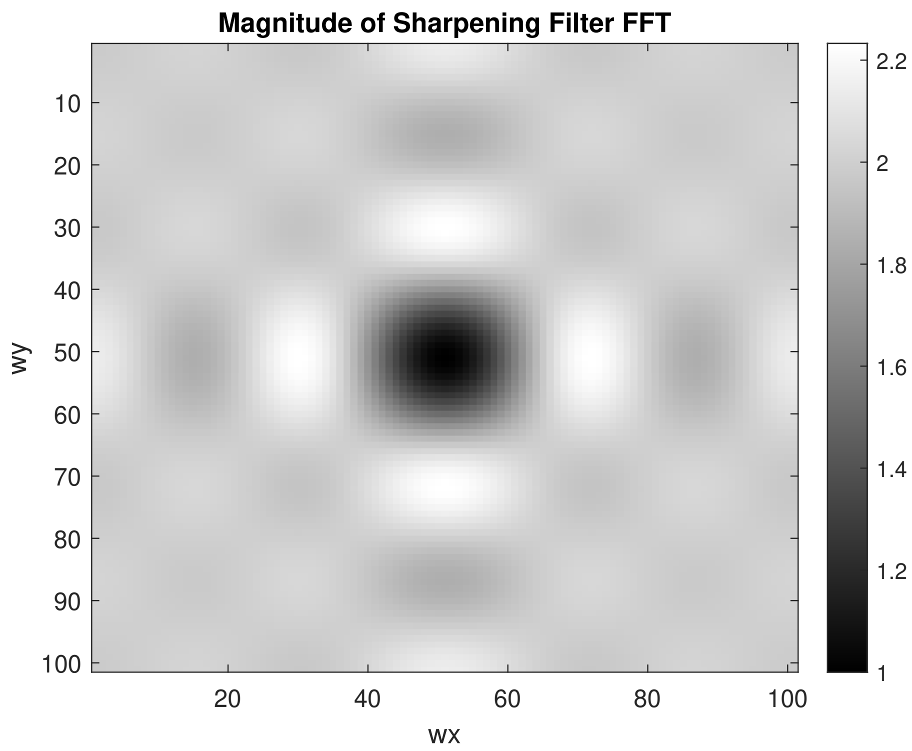


Figure 6: 2D Sharpening Filter

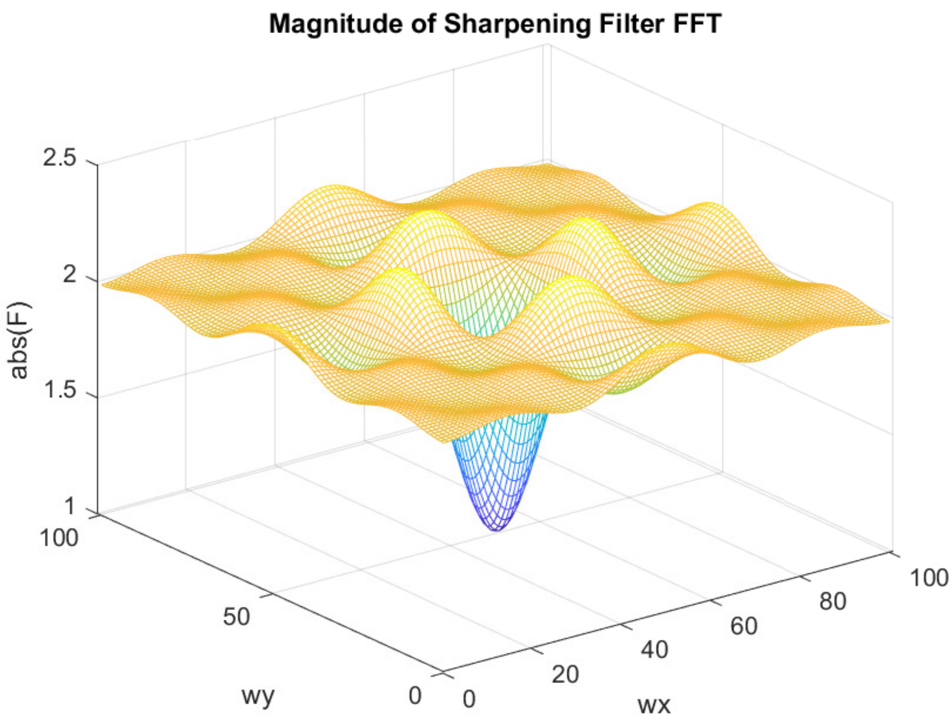


Figure 7: 3D Sharpening Filter

Figure 8: Magnitude of the transformed Sharpening Filter