UNIVERSITEIT GENT

FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

VAKGROEP TELECOMMUNICATIE EN INFORMATIEVERWERKING
ONDERZOEKSGROEP IMAGE PROCESSING AND INTERPRETATION

Multimedia – Labo 3

Gianni Allebosch, Martin Dimitrievski
{gianni.allebosch, martin.dimitrievski}@ugent.be

Master of Science in de industriële wetenschappen: elektronica-ICT

Campus Ardoyen,
BE - 9000 Gent

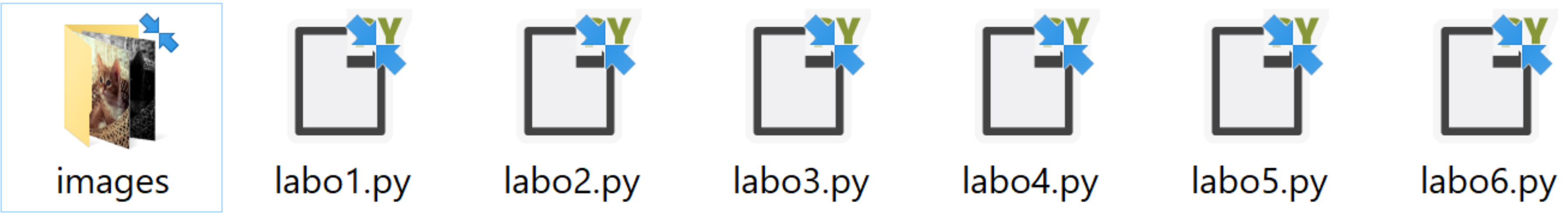
UNIVERSITEIT
GENT

TIPS & TRICKS

- **Keep your program user friendly!**
  - Provide a README.txt
  - Execute each exercise sequentially in 1 script
  - Make clear which exercise is currently being executed
  - Name your windows when showing images
  - If input is required from the user, make sure the instructions are clear

TIPS & TRICKS

- **Be sure the code is executable as is!**
  - We will not change your code!
  - Functions in comments will not be executed
  - Store all input images in a folder named “images” next to your code



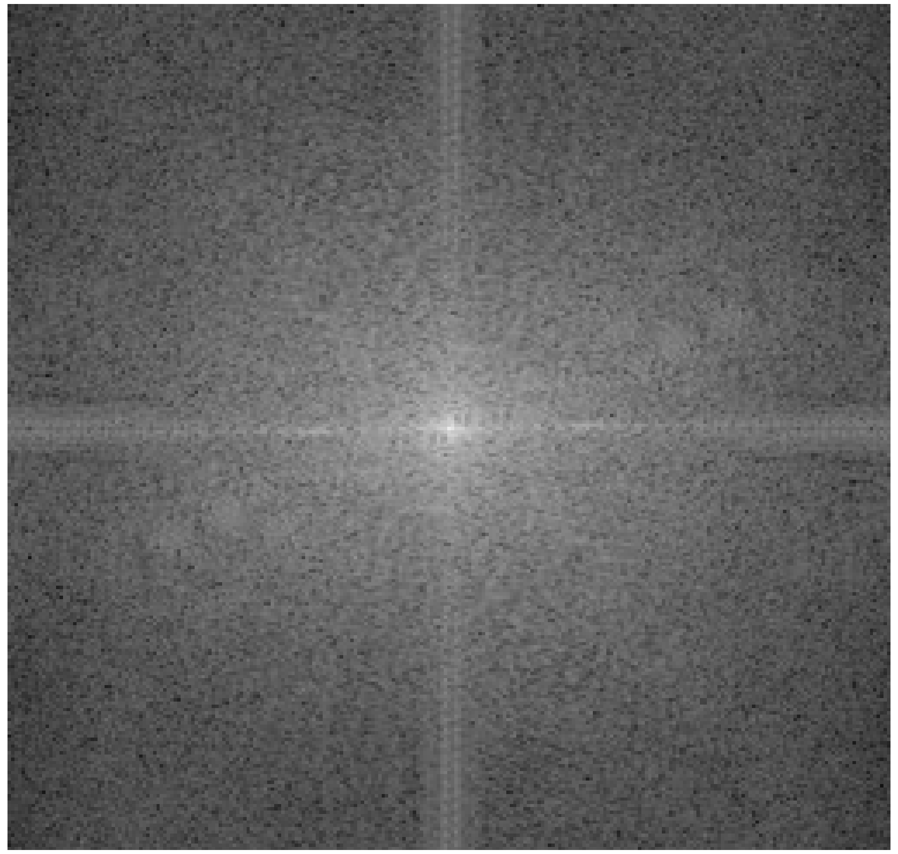
  - Do not use absolute paths but use relative paths (portability!):

    ✗ “C:/users/admin/multimedia/labo1/images/lena_color.jpg”

    ✔ “images/lena_color.jpg”

2D Discrete Fourier Transform

image

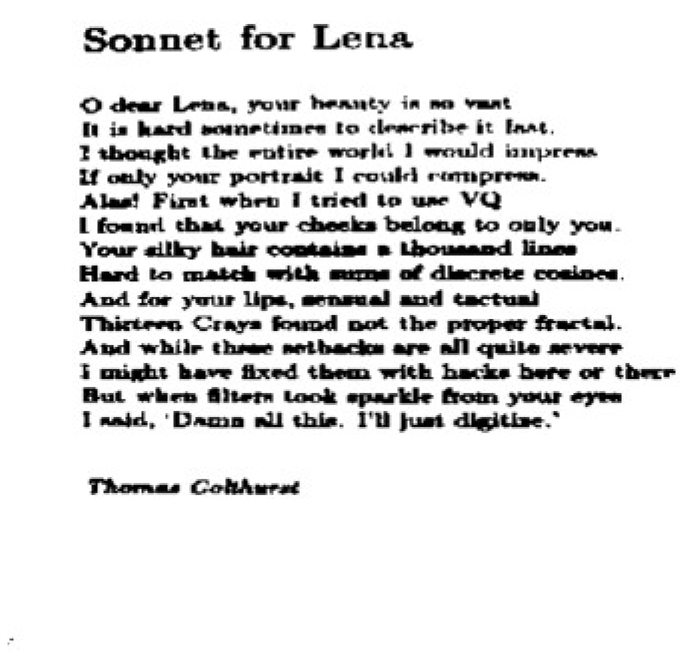


fft magnitude spectrum

Frequency domain → spatial changes/variations in intensity

- Low frequencies (close to the origin) = slowly changing variations in image e.g. a plane of homogeneous color
- High frequencies (away from the origin) = sudden changes in the image e.g. edges or noise
- DC-component (the origin) = average gray value of the image

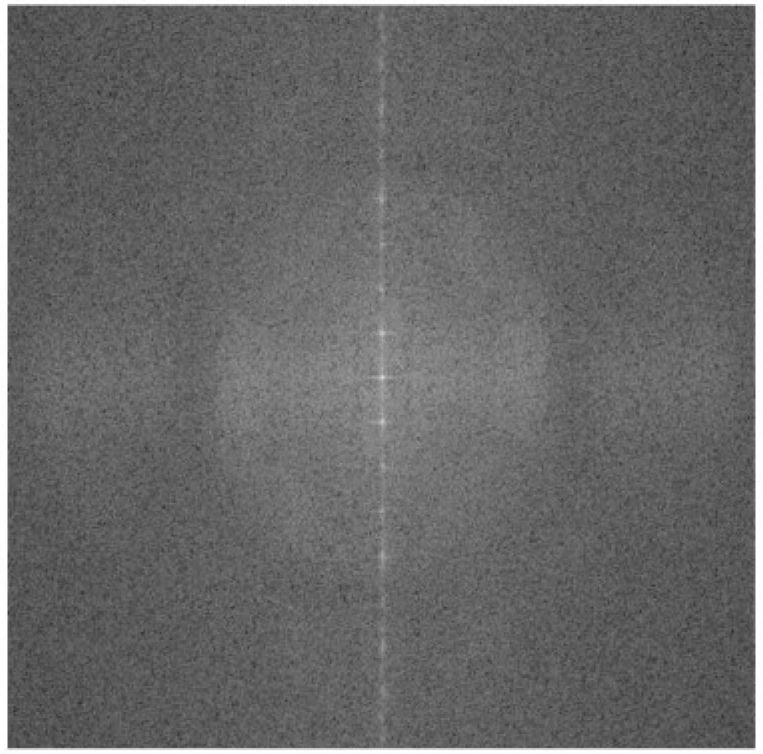
2D Discrete Fourier Transform

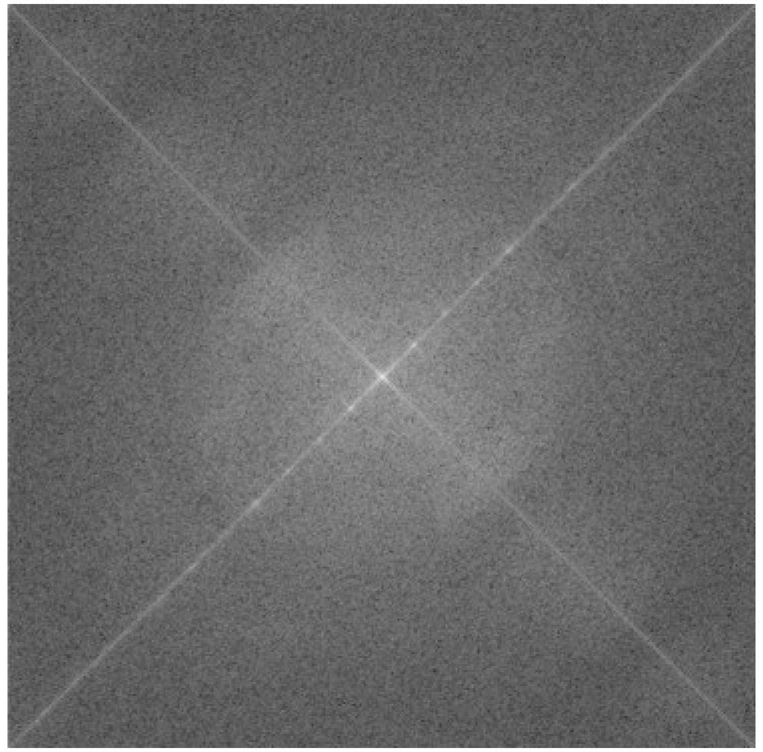


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

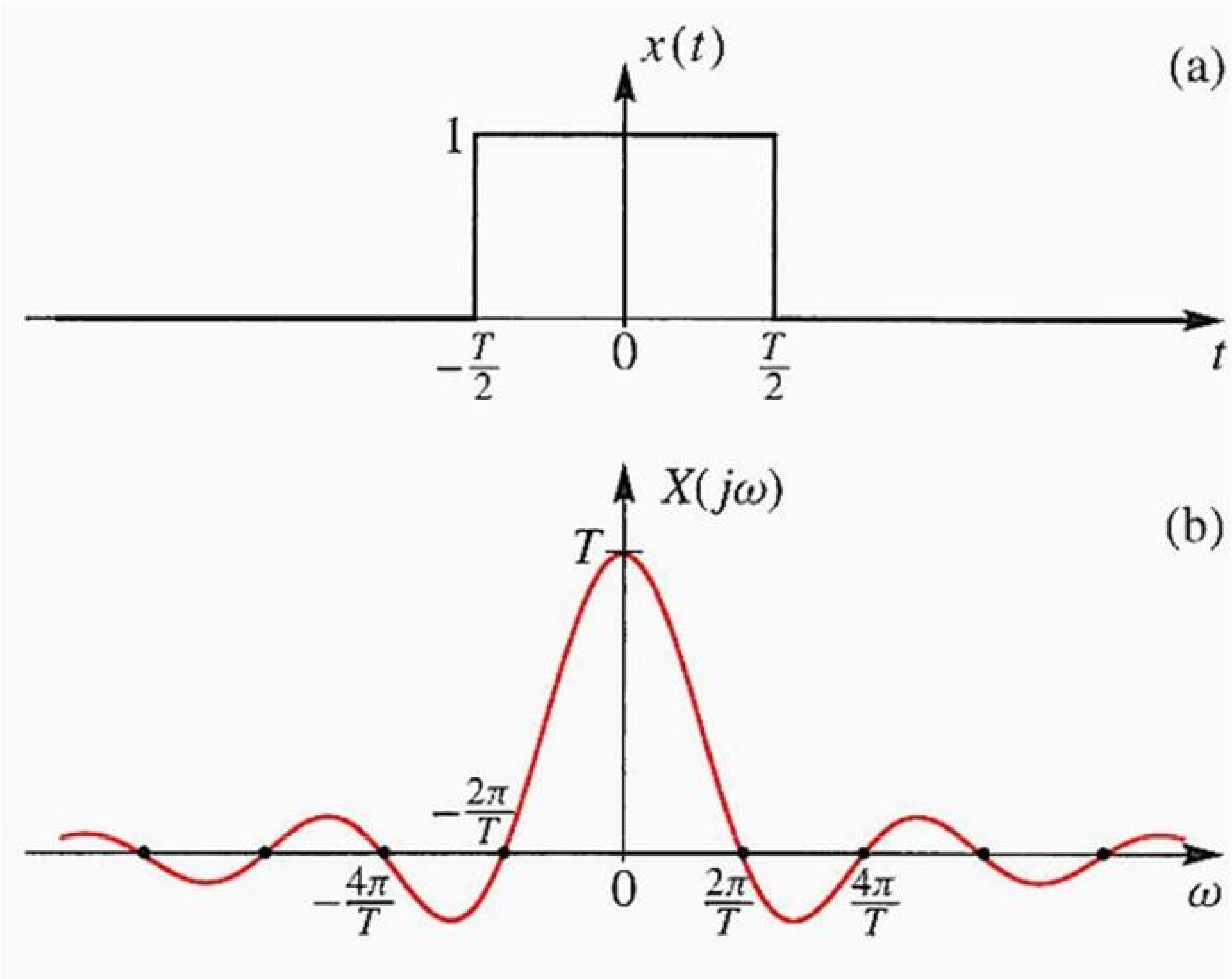
Thomas Colthurst





Frequency domain → spatial changes/variations in intensity
→ Major directions in images

2D Discrete Fourier Transform



Time domain

Rectanglular pulse with

T = pulse width

A = pulse amplitude

Fourier Transform

Frequency domain

Fading sinus (= ? Function) with

1/T = frequency

2π/T = angular frequency

Question – 2D Discrete Fourier Transform

Solve the question in your README.txt

According to what pattern are the zero points in the spectrum formed? Why?

2D Discrete Fourier Transform

2D Fourier → Fourier in x and y direction

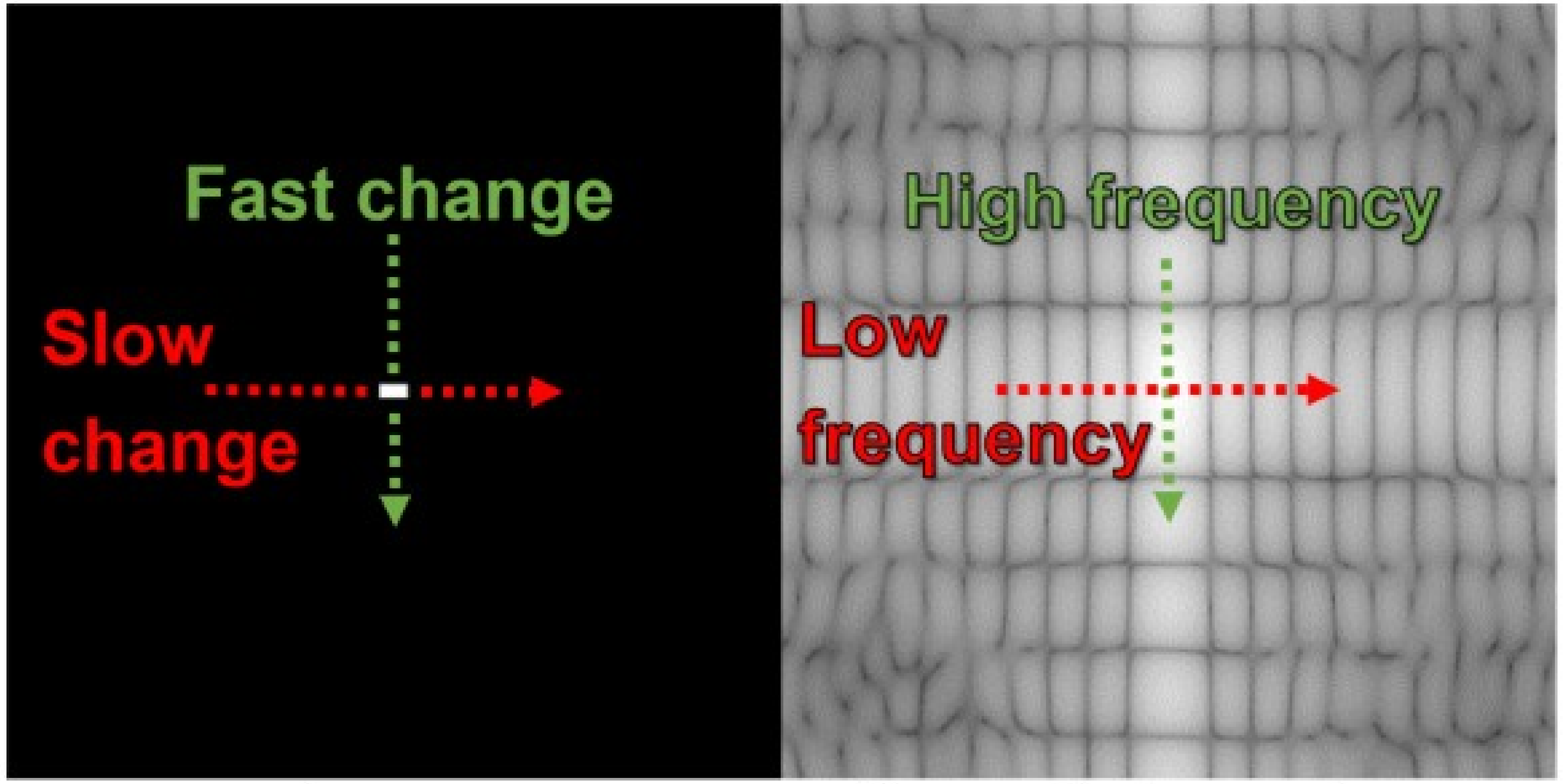
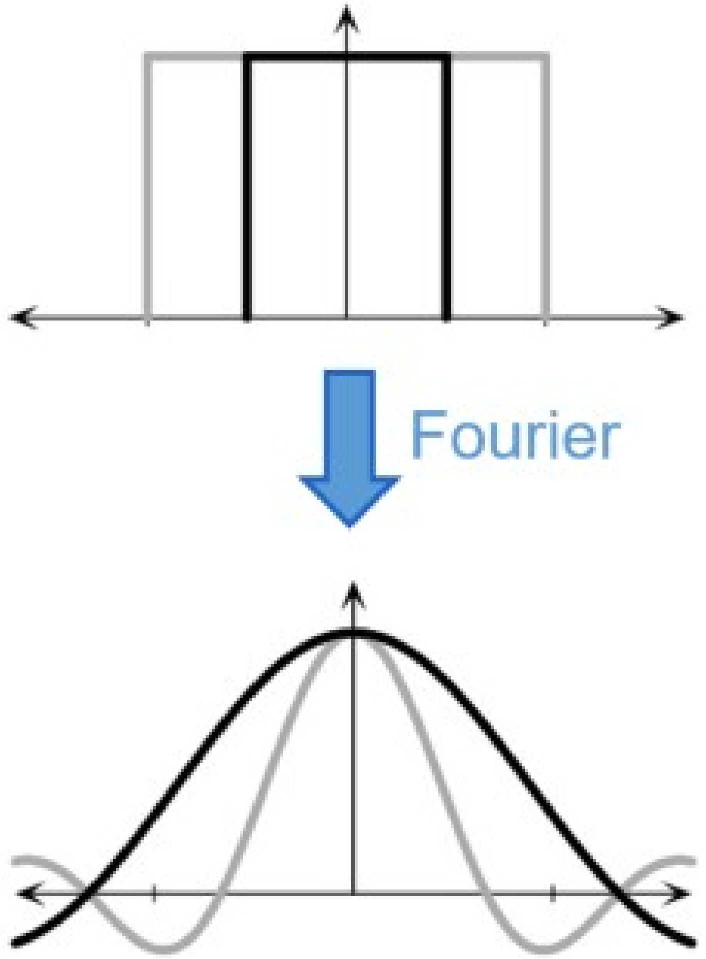


image fft magnitude spectrum



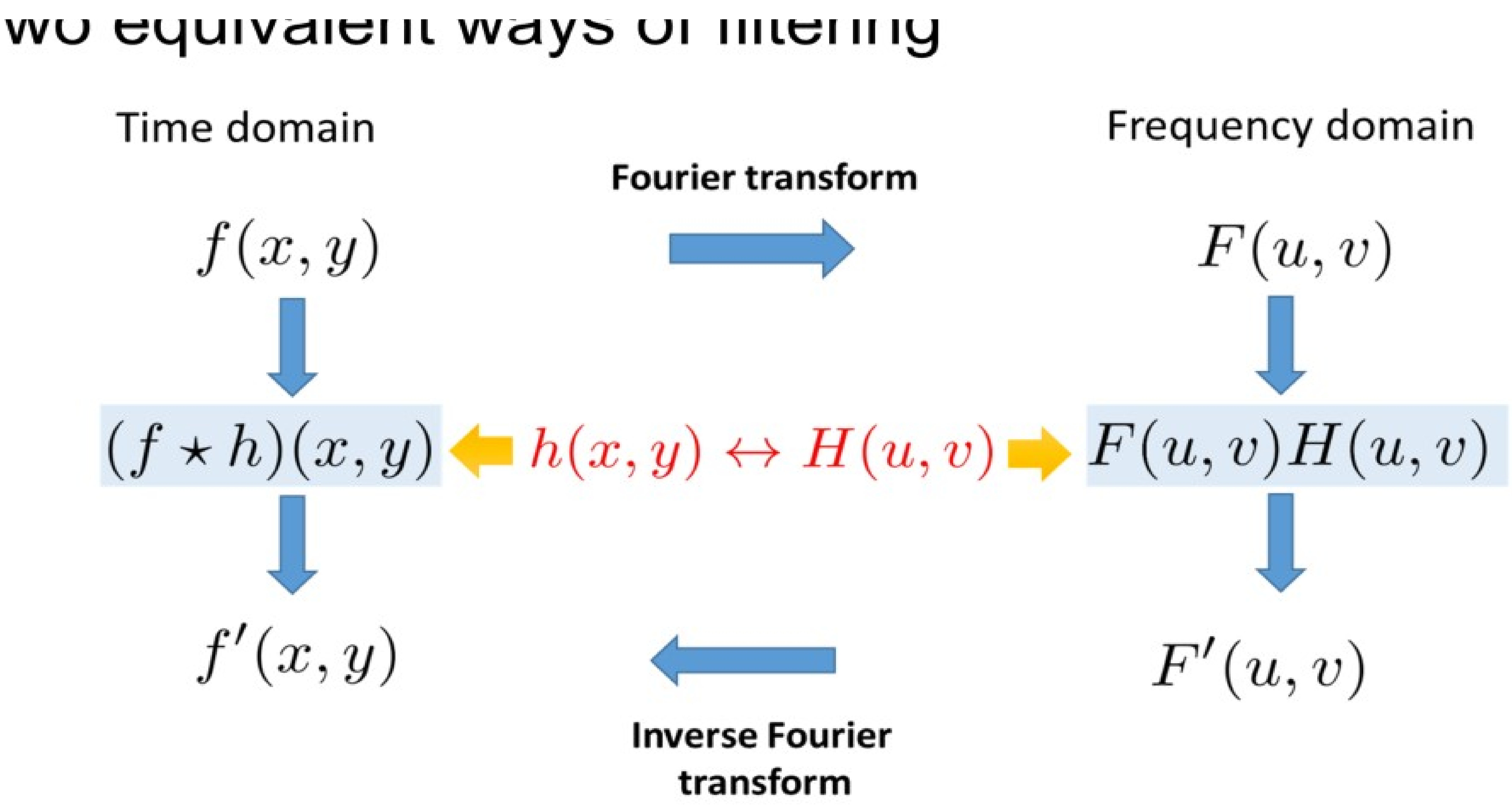
2D Discrete Fourier Transform

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

image = cv2.imread("disk.jpg", cv2.IMREAD_GRAYSCALE)  # grayscale images
fbeeld = np.fft.fft2(image)          # 2D Fourier transform
fshift = np.fft.fftshift(fbeeld)     # shift origin to center
plt.imshow(magnitude_spectrum(fshift), cmap='gray')  # show magnitude spectrum of fshift
plt.show()

ifshift = np.fft.ifftshift(fshift)     # shift origin to original place (is the same as fftshift but more readable)
ifbeeld = np.fft.ifft2(ifshift)        # inverse 2D Fourier transform
ifbeeld = ifbeeld.real                 # complex (because of rounding errors) to real values
plt.imshow(ifbeeld, cmap='gray')
plt.show()

def magnitude_spectrum(fshift):
    """ Returns magnitude spectrum of the 2D fourier transform for viewing purpose. """
    magnitude_spectrum = np.log(np.abs(fshift)+1)            # reduce large dynamic range
    magnitude_spectrum -= np.min(magnitude_spectrum)         # scale to range [0, 255]
    magnitude_spectrum *= 255./np.max(magnitude_spectrum)    # scale to range [0, 255]
    return magnitude_spectrum
```

UNIVERSITEIT GENT

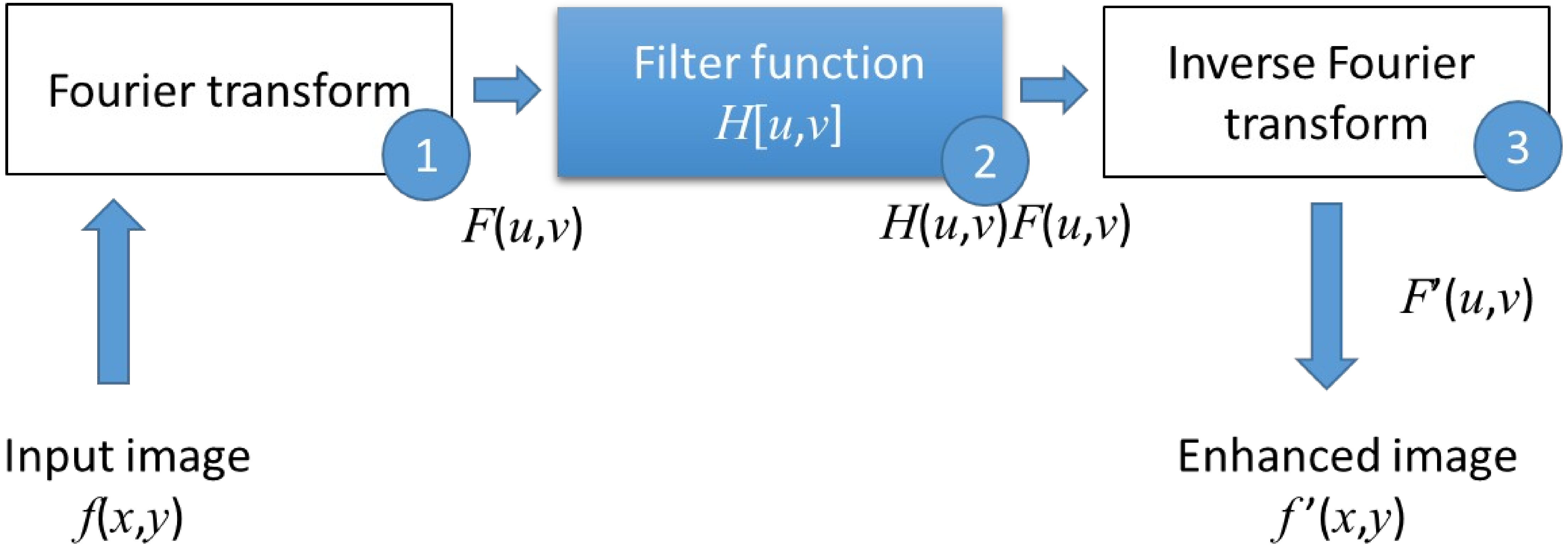
Filtering in the frequency domain

Two equivalent ways of filtering:



UNIVERSITEIT GENT

Filtering in the frequency domain



1. $F(u,v) = \mathcal{F}\left[f\right](u,v)$
2. $F'(u,v) = H(u,v)F(u,v)$
3. $f'(x,y) = \mathcal{F}^{-1}\left(F'\right)(x,y)$

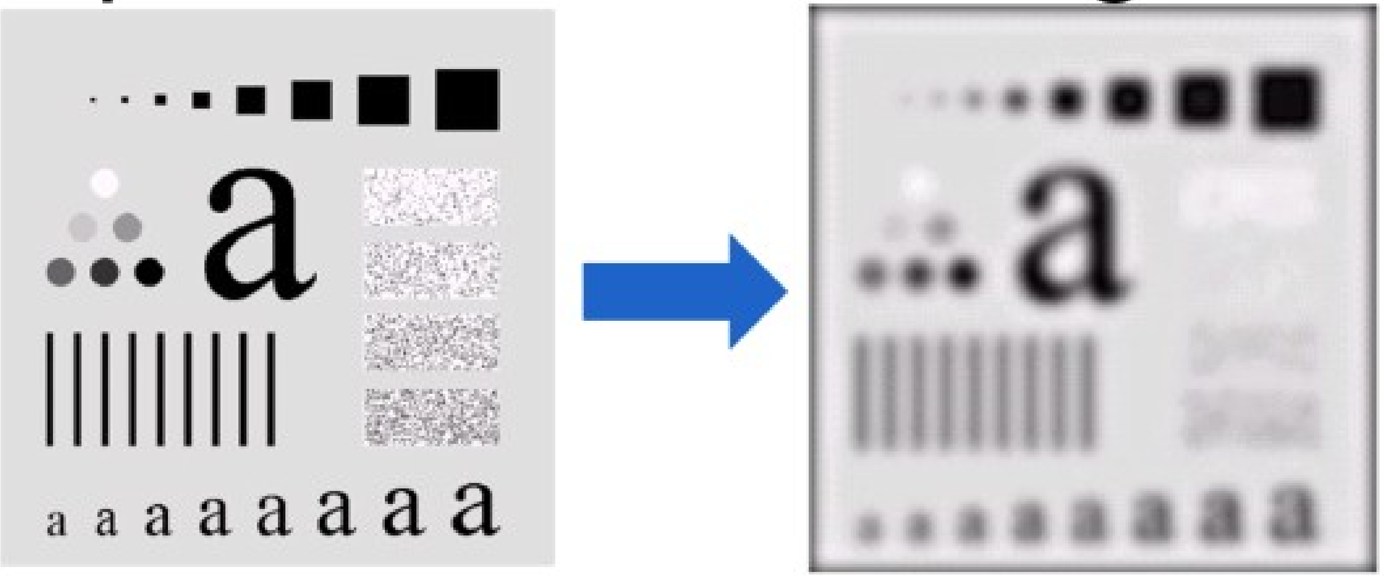
Notation conventions:

- Small letter: spatial domain
- Capital letter: frequency domain

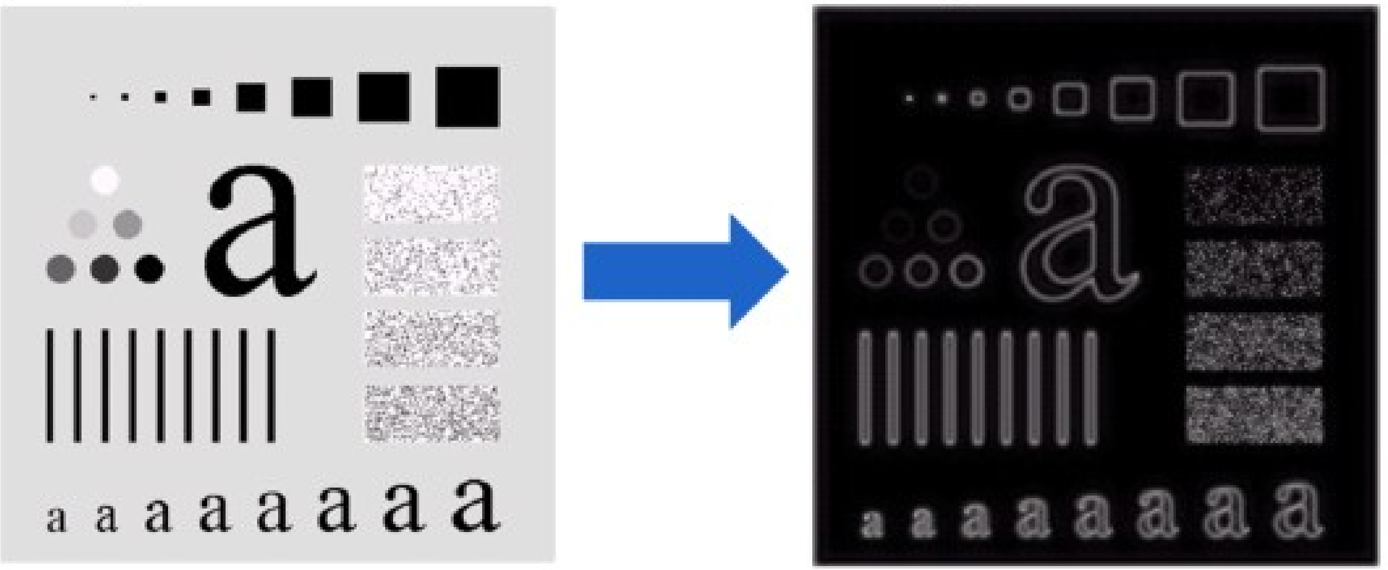
UNIVERSITEIT GENT

Filtering in the frequency domain

– Low-pass filters → blurring



– High-pass filters → edge detection



Lowpass Filters

Ideal lowpass filter

Butterworth lowpass

Gaussian lowpass filter

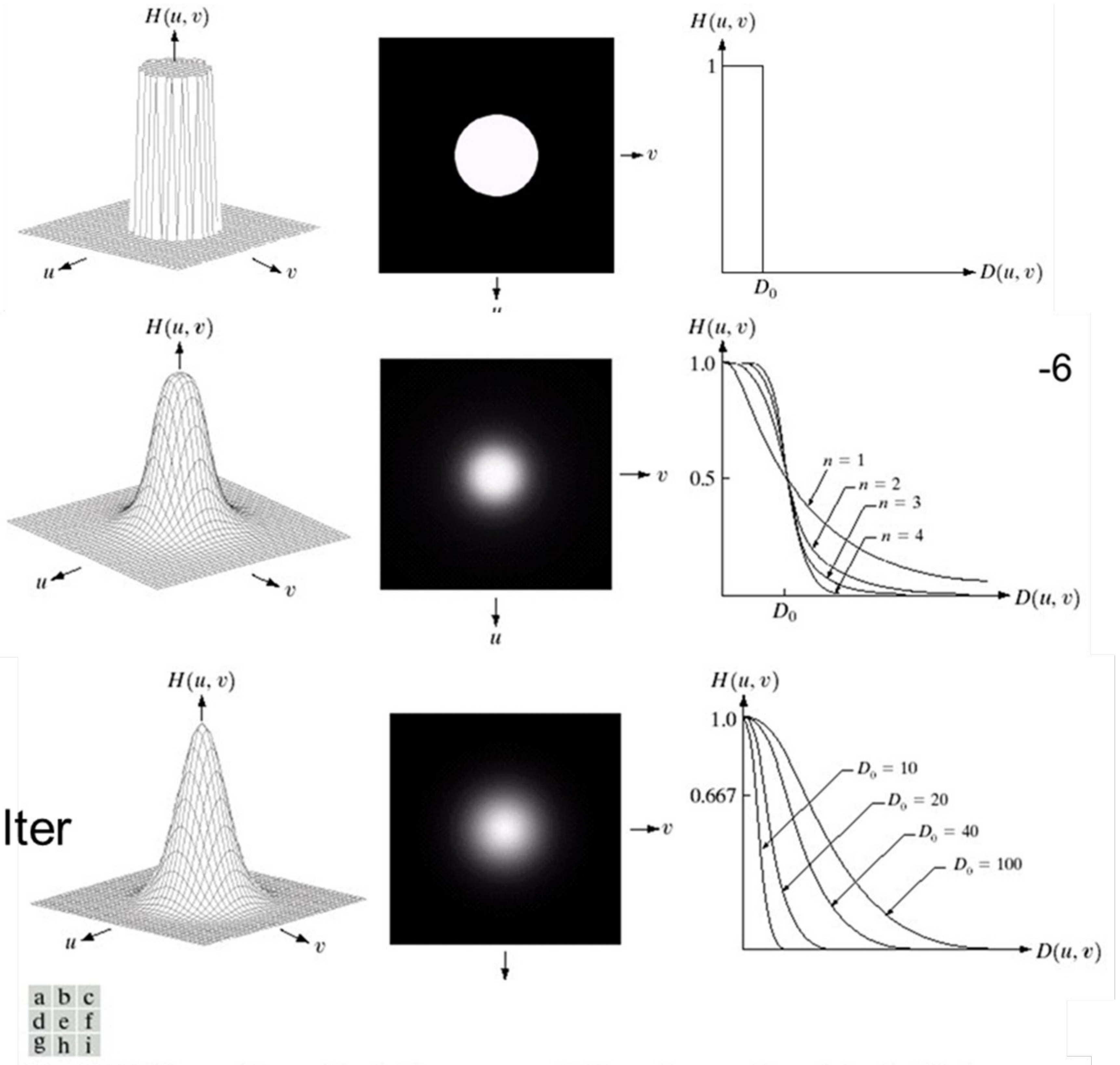


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal lowpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian lowpass filters.

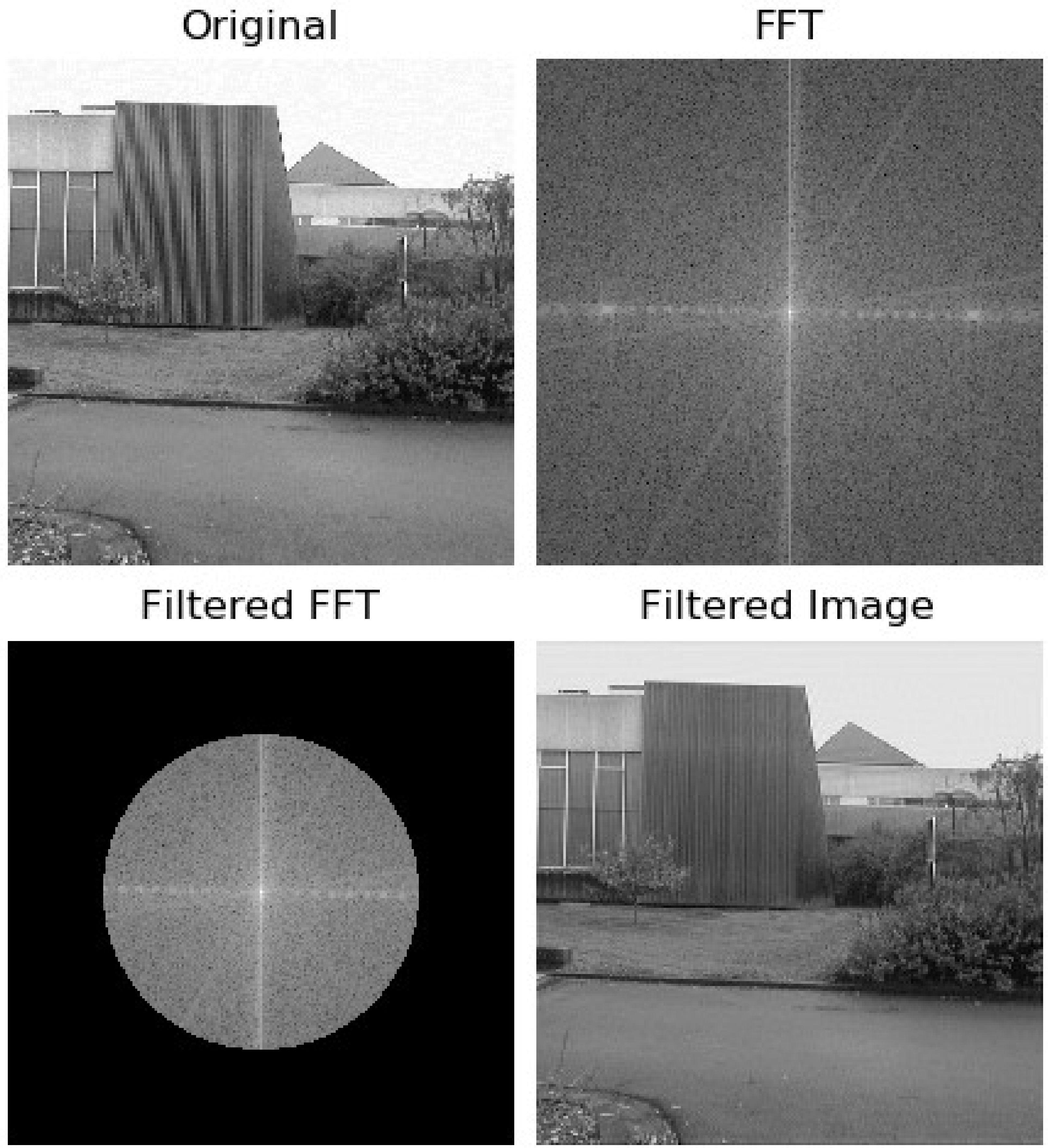
UNIVERSITEIT GENT

Exercise 1 – Ideal Lowpass Filter

Solve the questions in your README.txt

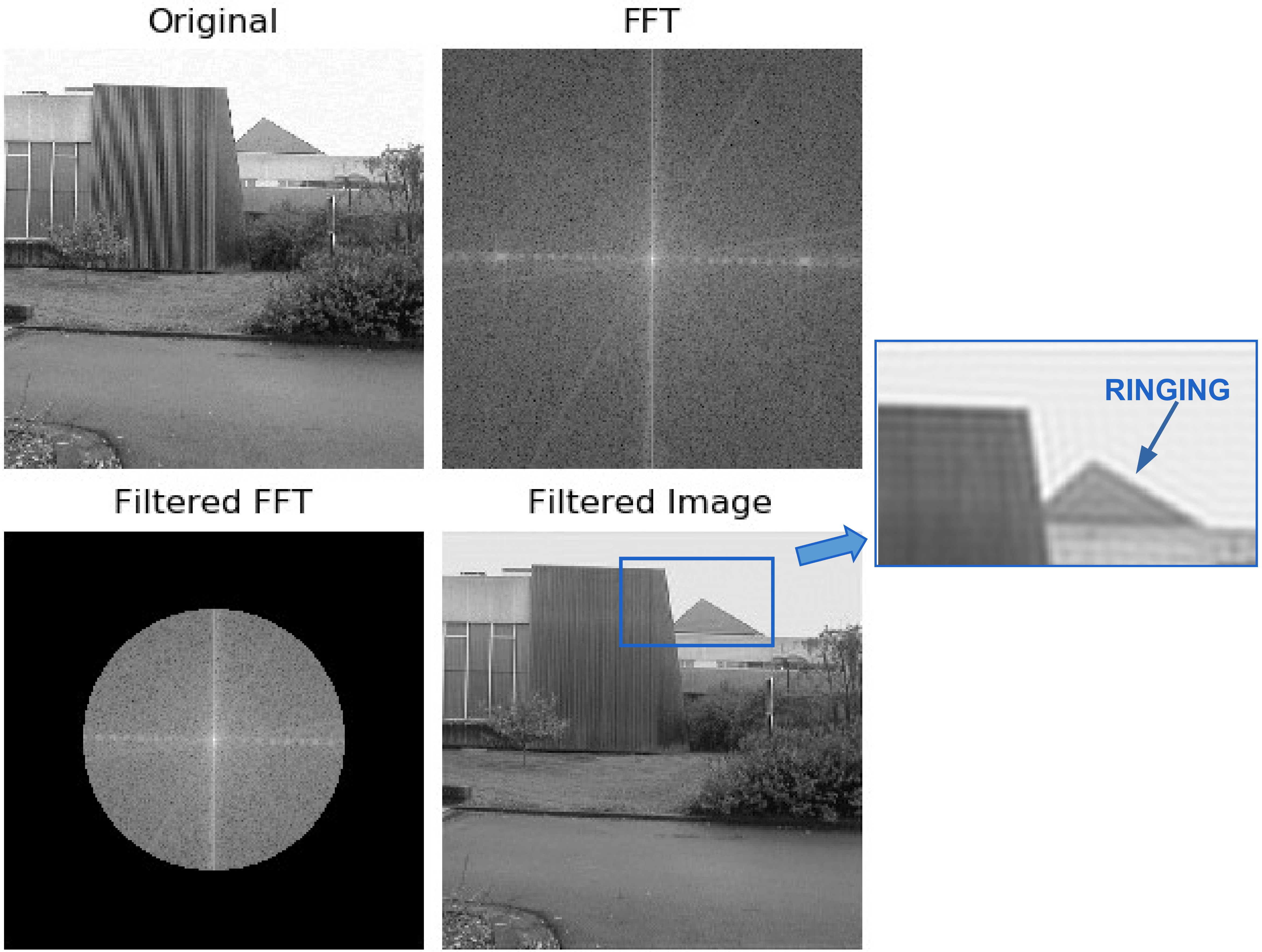
- What does the frequency content with highest magnitude represent?
- Discuss what happens when the radius of the circular filter varies.
- Discuss the effects of the ideal lowpass filter and ringing.

Exercise 1 – Ideal Lowpass Filter

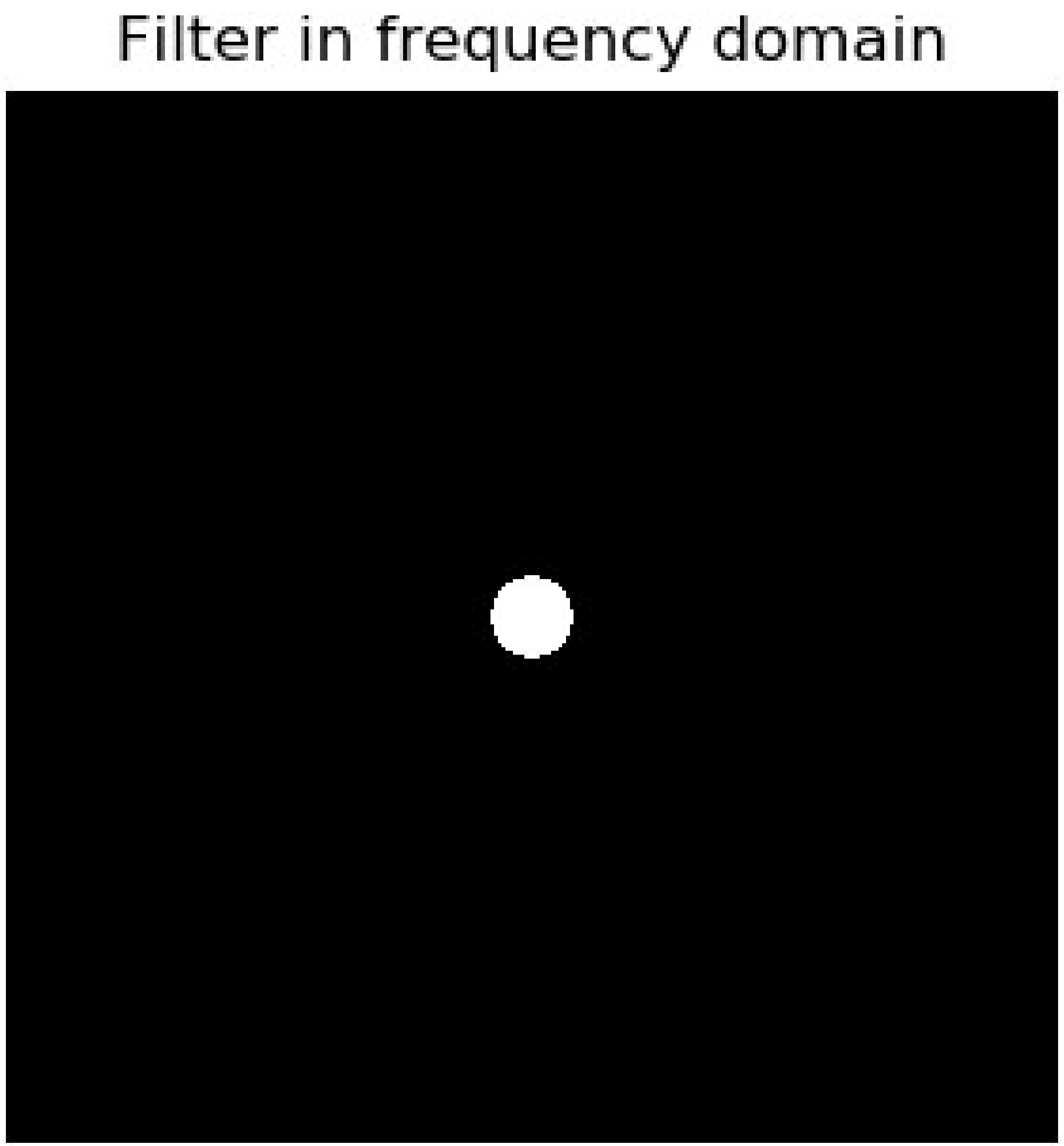


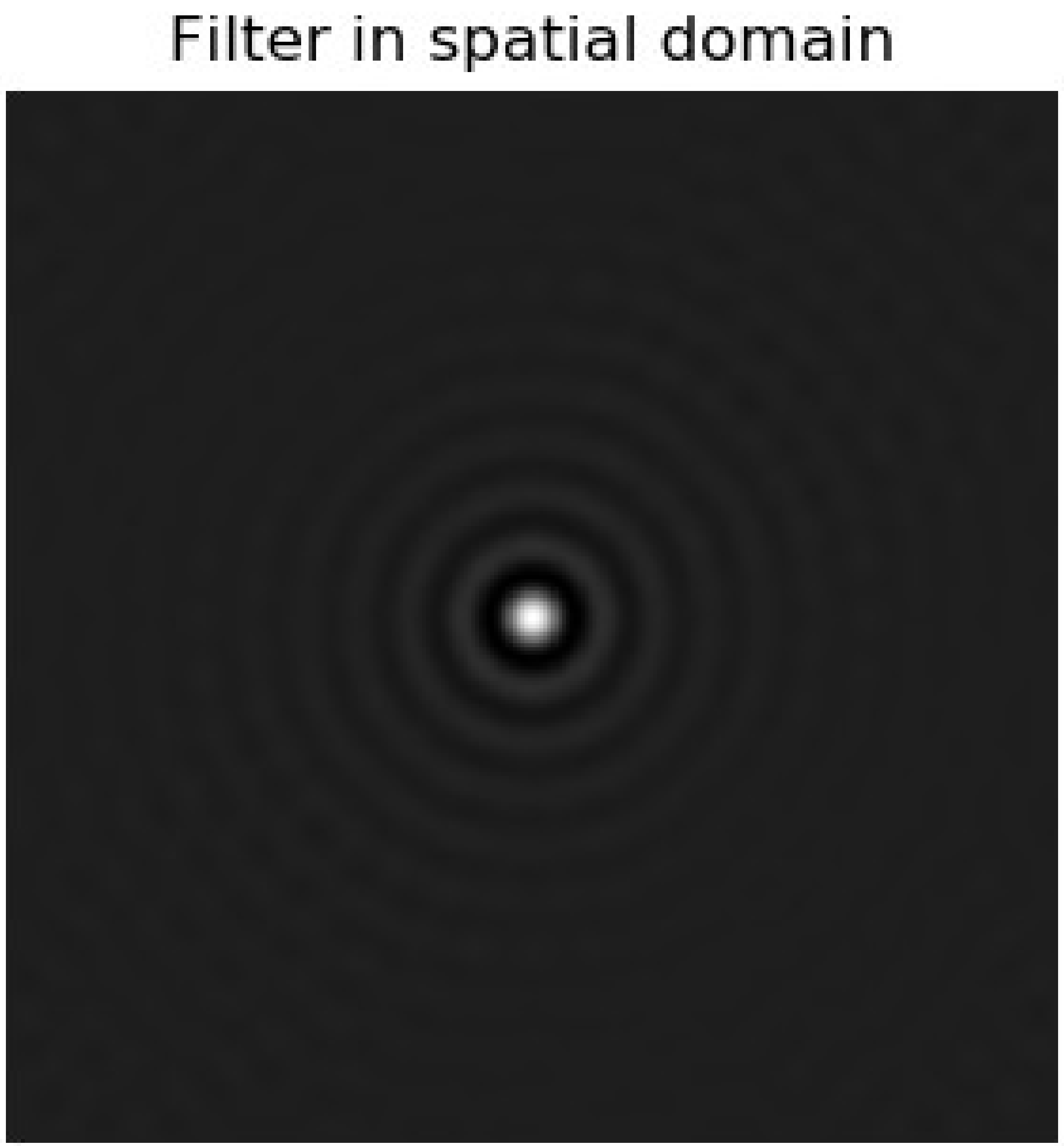
UNIVERSITEIT GENT

Exercise 1 – Ideal Lowpass Filter



Exercise 1 – Ideal Lowpass Filter



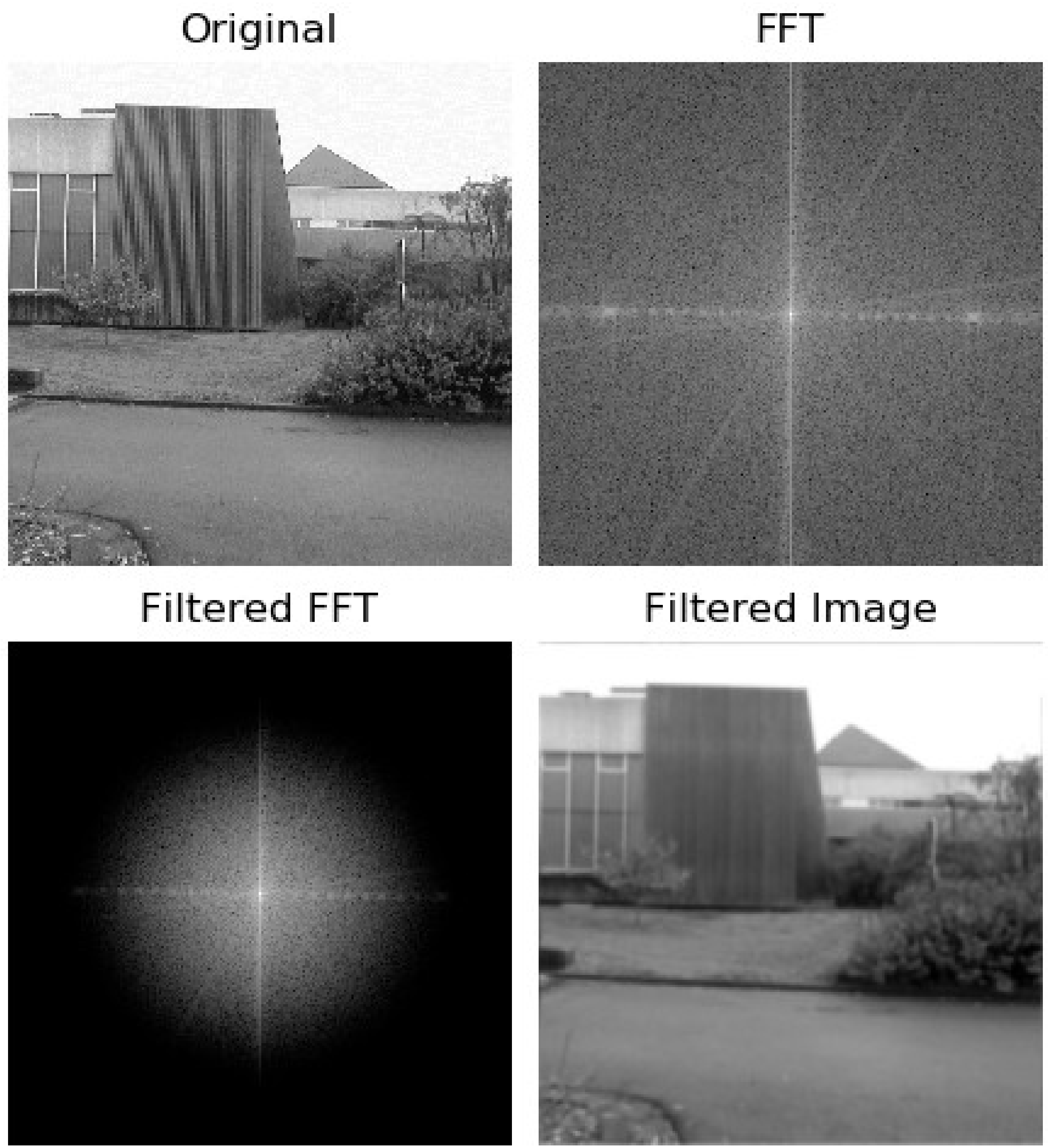


Exercise 2 – Gaussian Lowpass Filter

Solve the questions in your README.txt

- Discuss the advantages and disadvantages of the Gaussian lowpass filter.
- Discuss what happens when D_0 varies.

Exercise 2 – Gaussian Lowpass Filter



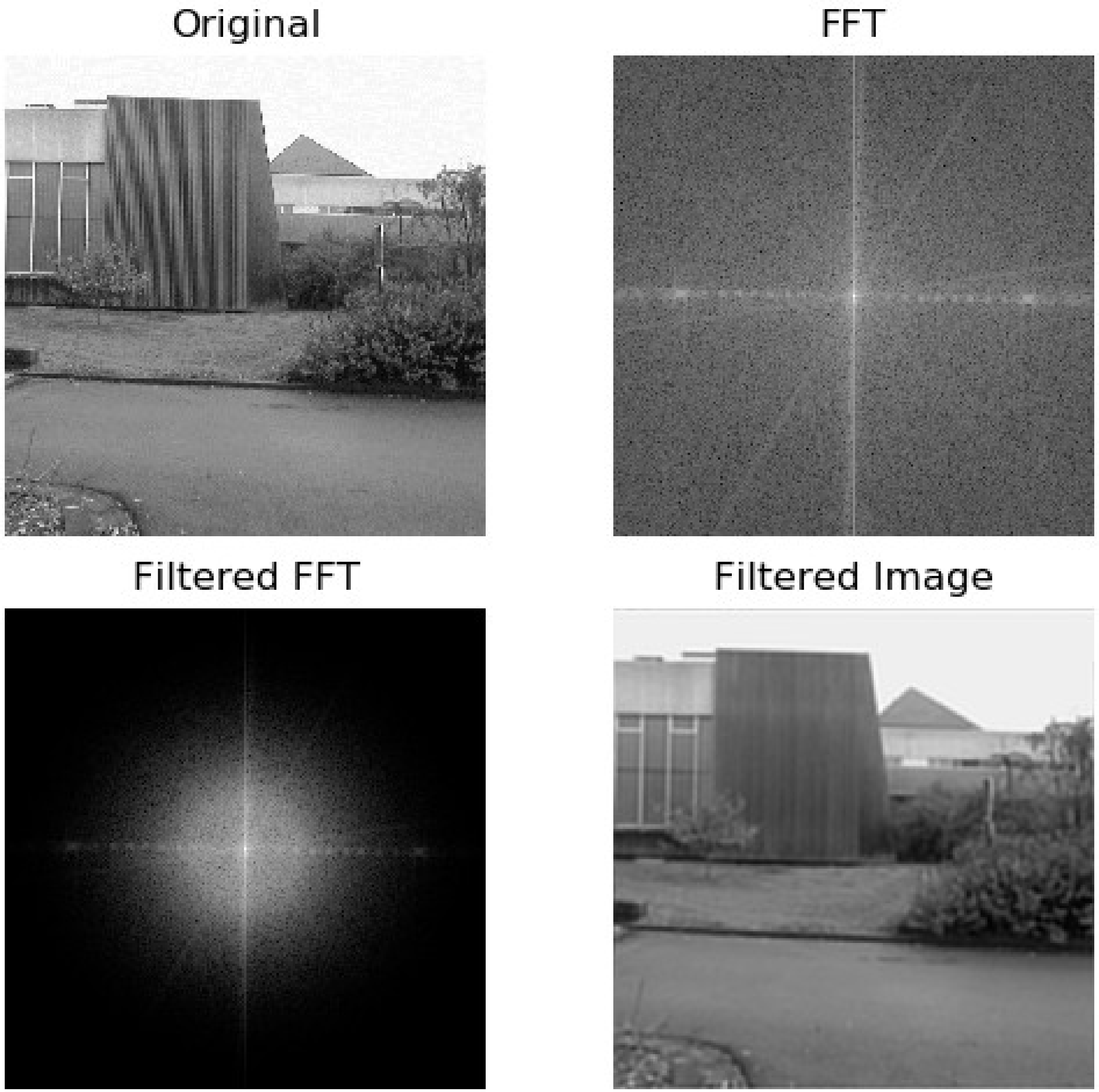
UNIVERSITEIT GENT

Exercise 3 – Butterworth Lowpass Filter

Solve the question in your README.txt

Discuss how the Butterworth lowpass filter solves some disadvantages of the Gaussian lowpass filter.

Exercise 3 – Butterworth Lowpass Filter



UNIVERSITEIT
GENT

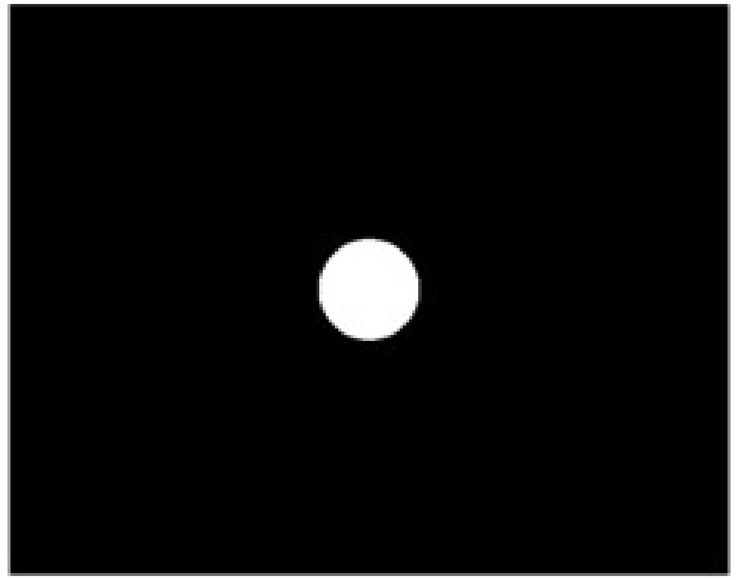
Highpass Filters

– Highpass filter:

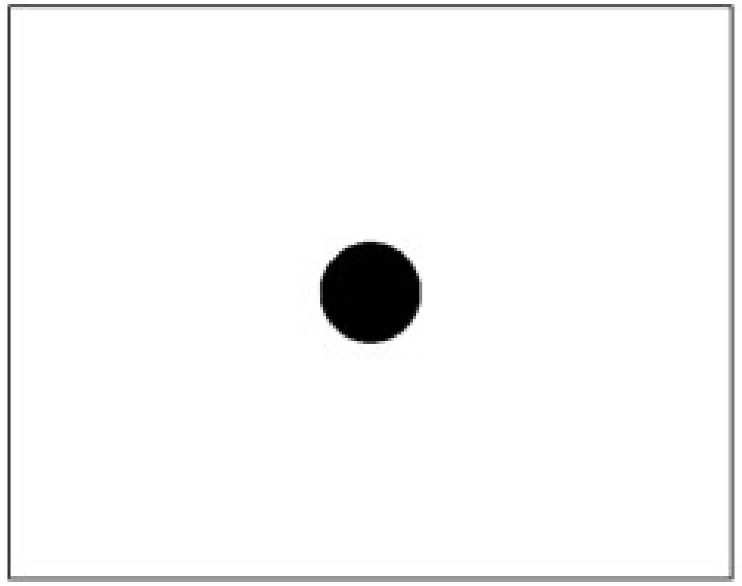
$$H_{HP} = 1 - H_{LP}$$

– High-frequency emphasis filter:

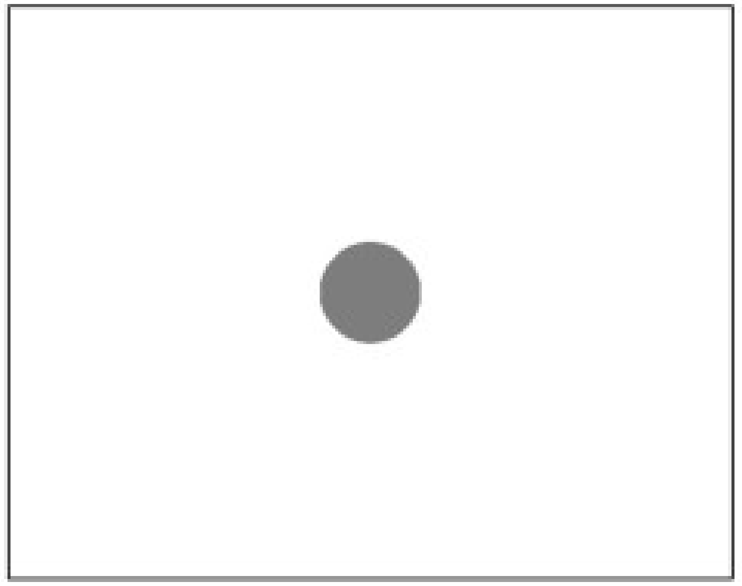
$$H_{HFE} = a + bH_{HP} \qquad with\ a < b$$



$H_{LP,ideal}$



$H_{HP,ideal}$



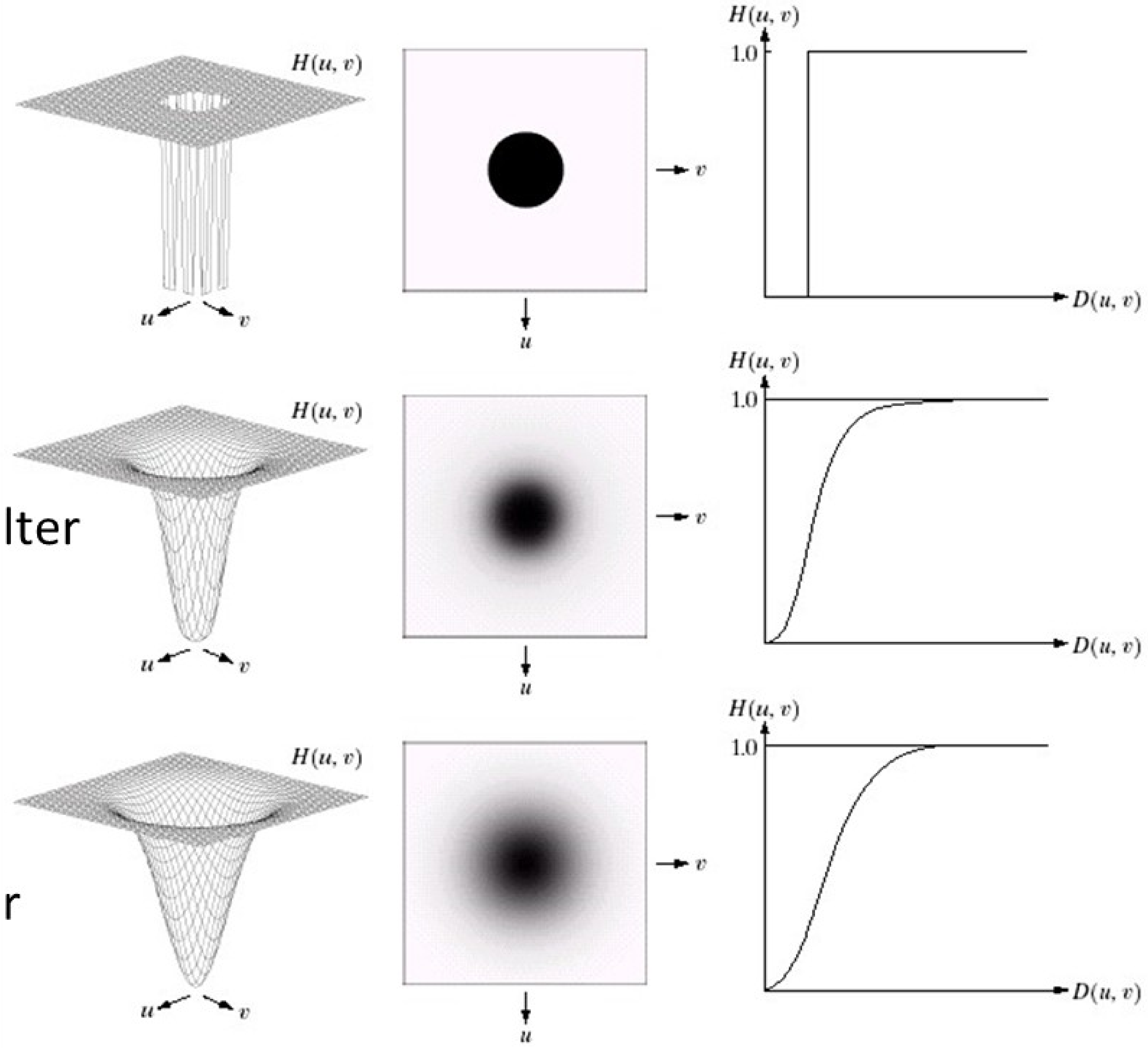
$H_{HFE,ideal}$

Highpass Filters

Ideal highpass filter

Butterworth highpass filter

Gaussian highpass filter

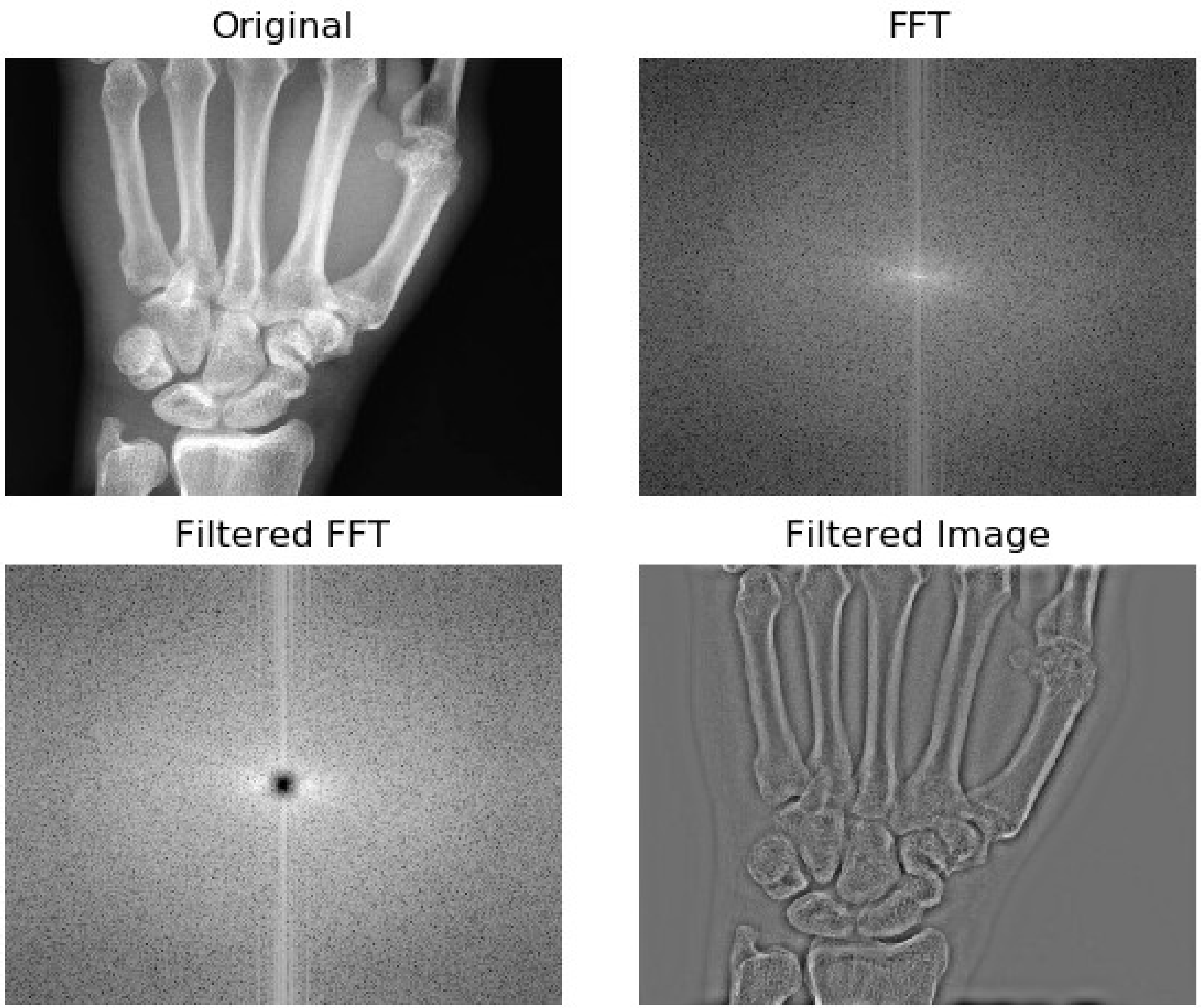


a b c
d e f
g h i

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

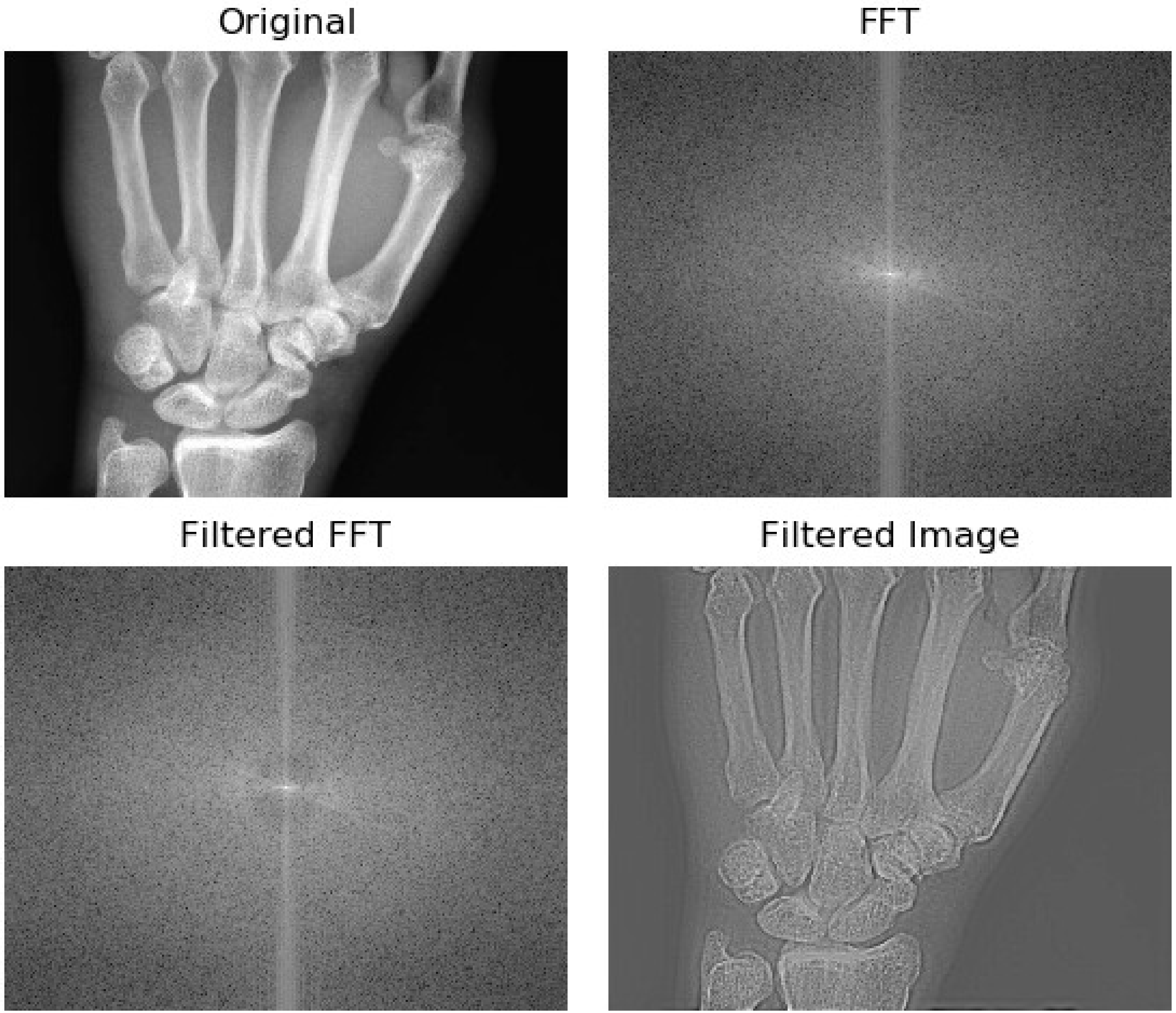
UNIVERSITEIT GENT

Exercise 4 – Butterworth Highpass Filter



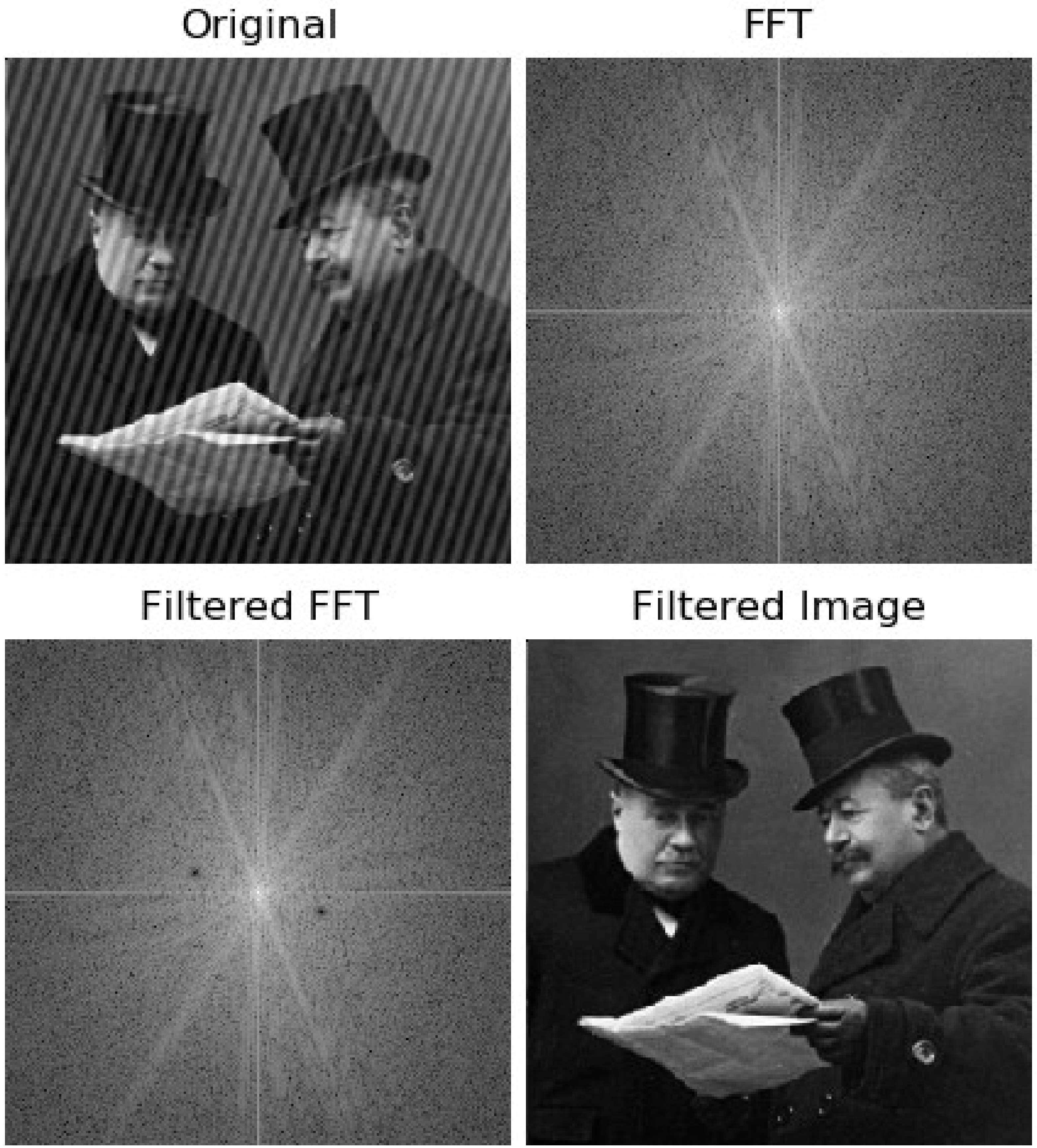
UNIVERSITEIT GENT

Exercise 4 - Butterworth High Frequency Emphasis Filter



UNIVERSITEIT GENT

Exercise 5 – Butterworth Notch Filter



TIPS & TRICKS

- **Deliver your working code before next session**
  - *Opdrachten* on Ufora
  - 1 python script named **labo#.py**
  - README.txt (see example)
  - **output.zip** with every output image

TIPS & TRICKS

- **Questions or need help? Contact us at:**

Gianni.Allebosch@UGent.be AND Martin.Dimitrievski@UGent.be

GOOD LUCK!