Recap of the previous lesson

Università di Catania

Luca Guarnera

Convolutions

Why "convolutional" filters?

- Linear and translation invariant filters are also called **convolutional filters**.
- We need to study the **convolution operation** to better understand how a filter can be computed.
- In addition, convolution is an extremely important phenomenon for all kinds of signal processing and for describing many physical events.

Convolution: properties

- To indicate the convolution operation, we use the notation

$$h = f \otimes g$$

- The convolution is commutative

$$f \otimes g = g \otimes f$$

- The convolution is associative

$$(f \otimes g) \otimes h = f \otimes (g \otimes h)$$

In the finite case (1)

- If kernel f has size ***kxh***, the formula should be rewritten as follows.

$$h_{m,n} = \sum_{i=-\lfloor k/2 \rfloor}^{\lceil k/2 \rceil - 1} \sum_{j=-\lfloor h/2 \rfloor}^{\lceil h/2 \rceil - 1} \left(f_{i,j} * g_{m+i,n+j} \right)$$

	-1	*0*	*1*
-1	**a**	**b**	**c**
0	**d**	**e**	**f**
1	**g**	**h**	**i**

- If the kernel indexes are arranged so that the coordinate point (0,0) is at the central position.

Nel caso finito (2)

- If kernel f has size ***kxh***, the formula should be rewritten as follows.

$$h_{m,n} = \sum_{i=1,\, j=1}^{k,h} f_{i,j} * g_{m+(i-k+\lfloor k/2 \rfloor),\, n+(j-h+\lfloor h/2 \rfloor)}$$

	1	2	3
1	a	b	c
2	d	e	f
3	g	h	i

- If the kernel indexes are arranged starting from 1 until h or k.

Example

	-1	0	1
-1	2	1	2
0	1	2	1
1	2	1	2

	1	2	3	4	5
1	1	2	2	3	1
2	3	2	2	1	4
3	2	5	2	7	1
4	9	0	1	1	2
5	3	1	2	4	1

	1	2	3	4	5
1					
2					
3					
4					
5					

Example

	-1	0	1
-1	2	1	2
0	1	2	1
1	2	1	2

	1	2	3	4	5
1	1	2	2	3	1
2	3	2	2	1	4
3	2	5	2	7	1
4	9	0	1	1	2
5	3	1	2	4	1

	1	2	3	4	5
1					
2		30			
3					
4					
5					

	-1	0	1
-1	2	1	2
0	1	2	1
1	2	1	2

	1	2	3	4	5
1	1	2	2	3	1
2	3	2	2	1	4
3	2	5	2	7	1
4	9	0	1	1	2
5	3	1	2	4	1

	1	2	3	4	5
1					
2		30	45	30	
3					
4					
5					

	-1	0	1
-1	2	1	2
0	1	2	1
1	2	1	2

	1	2	3	4	5
1	1	2	2	3	1
2	3	2	2	1	4
3	2	5	2	7	1
4	9	0	1	1	2
5	3	1	2	4	1

	1	2	3	4	5
1					
2		30	45	30	
3		46	27	37	
4					
5					

	-1	0	1
-1	2	1	2
0	1	2	1
1	2	1	2

	1	2	3	4	5
1	1	2	2	3	1
2	3	2	2	1	4
3	2	5	2	7	1
4	9	0	1	1	2
5	3	1	2	4	1

	1	2	3	4	5
1					
2		30	45	30	
3		46	27	37	
4		34	41	28	
5					

In the implementation

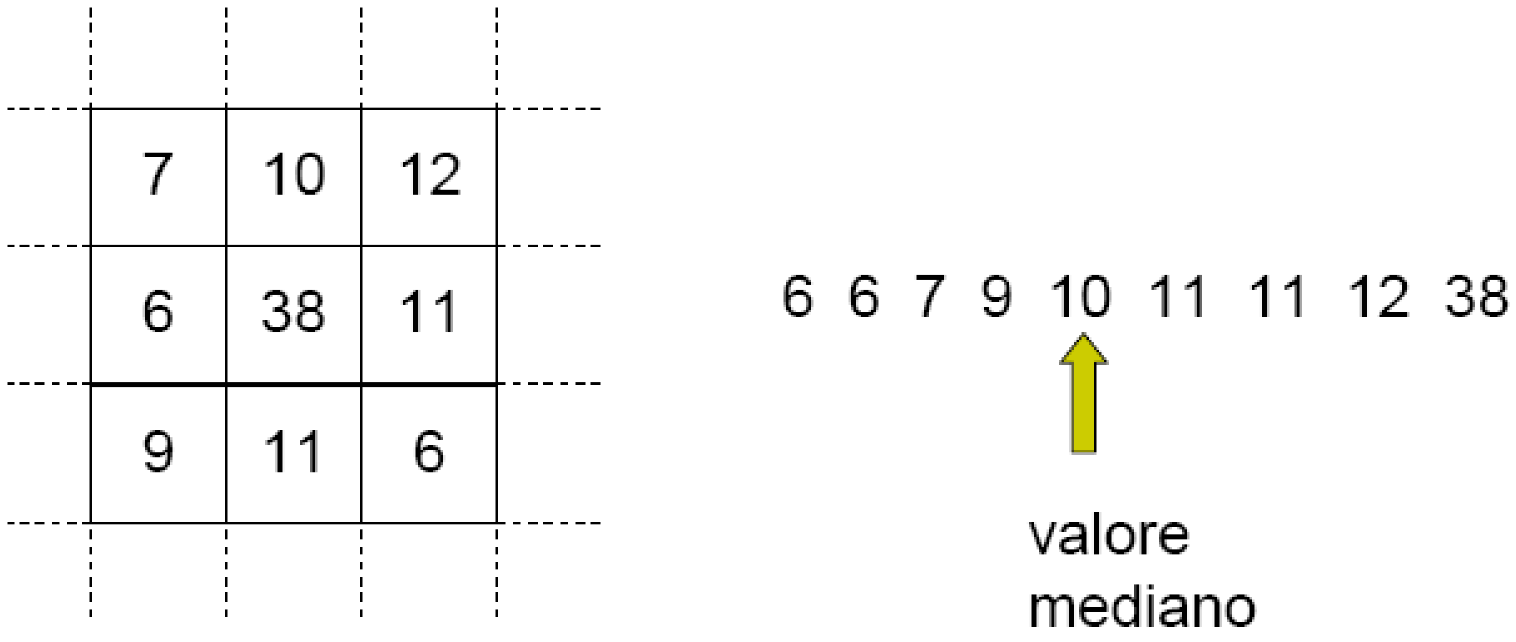
One problem is with edges: how to do convolution and filtering at edges?

POSSIBLE SOLUTIONS:

a) Filter only the central areas of the image
b) Assume that all around the image there is 0
c) Assume a "toroidal" topology: when you "overflow to the right" you indent to the left, when you "overflow" to the bottom you indent to the top and vice versa;
d) Add a row at the beginning equal to the previous rows, a row at the end equal to the last row, a column at the beginning equal to the starting column, and a column at the end equal to the ending column.

Median

- It is a nonlinear filter that outputs the median value of the pixel's neighborhood.



Median filter

Minimum filter

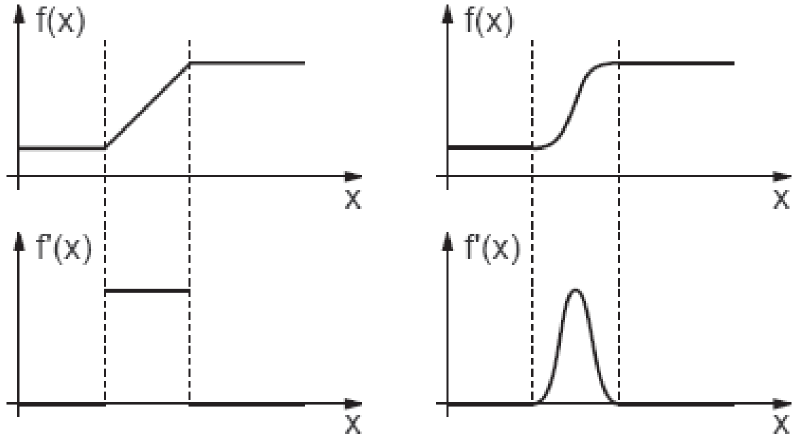
Maximum filter

Edge extraction

- Local operators help us extract edges from an image.
- **Edges are defined as local luminance discontinuities.**
- **Edge detectors provide images in which luminance variations are preserved and all other information is removed.**

Edge detectors based on the first derivative

- If I have a one-dimensional signal and calculate the first derivative, I find that the **edges are the correspondences of the** **<u>maxima of the derivative</u>**.

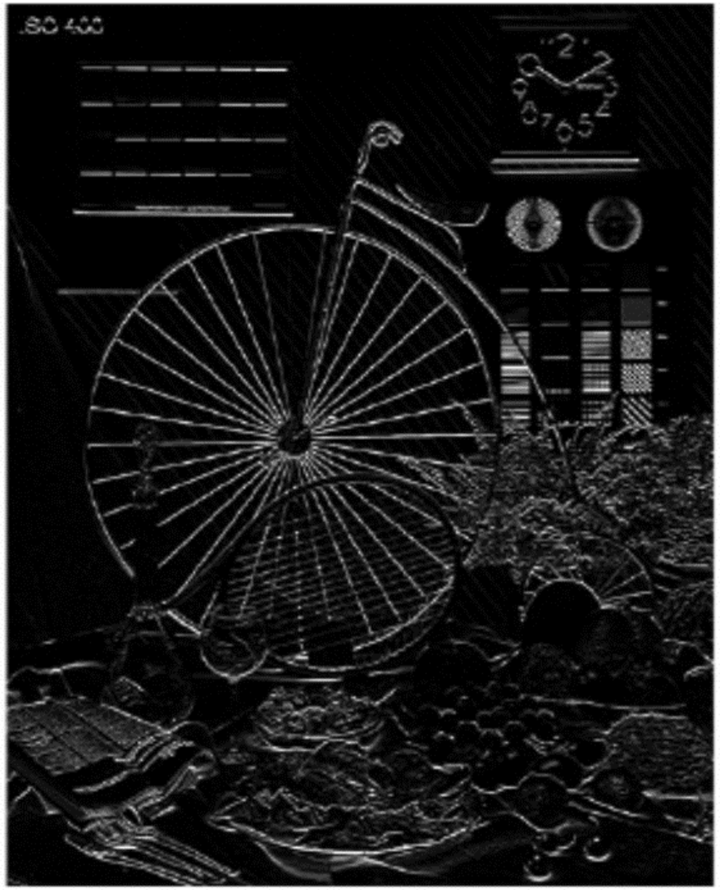


- Then the filters have to calculate the derivative in the x-direction that in the y-direction and then combine them together.

Sobel x

-1	-2	-1
0	0	0
1	2	1

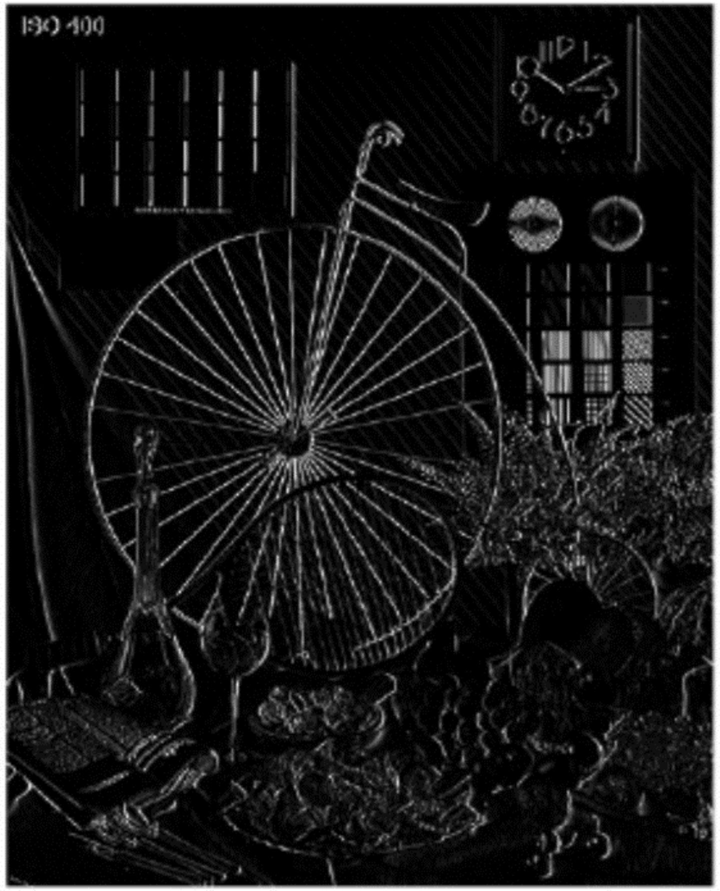
Filtro x-Sobel



Sobel y

-1	0	1
-2	0	2
-1	0	1

Filtro y-Sobel



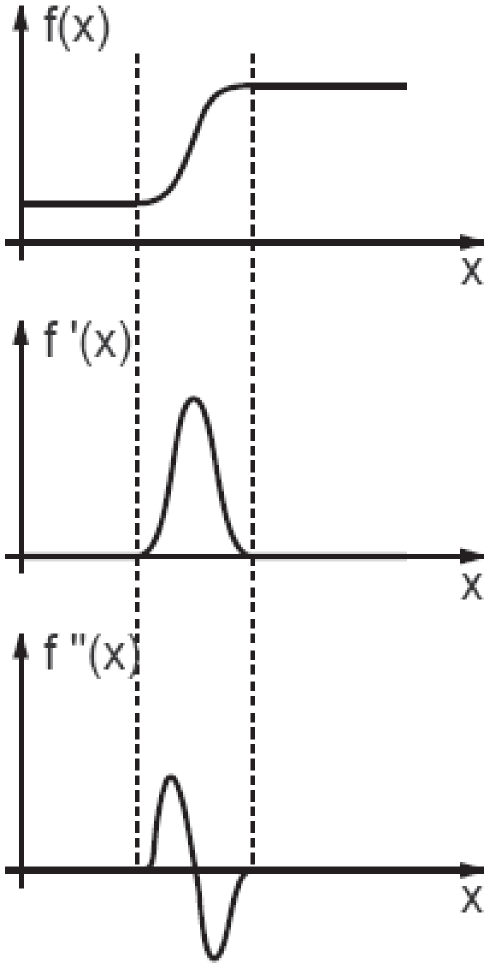
Better results...

- **They are obtained with more sophisticated (nonlinear) algorithms for calculating the magnitude of the gradient (sum of the square of the response of a horizontal edge finder and the square of the response of a vertical edge finder)**

- **They are obtained with more "intelligent" strategies (Canny's algorithm, fuzzy algorithms, backtracking techniques, etc.)**

Edge detectors based on the second derivative

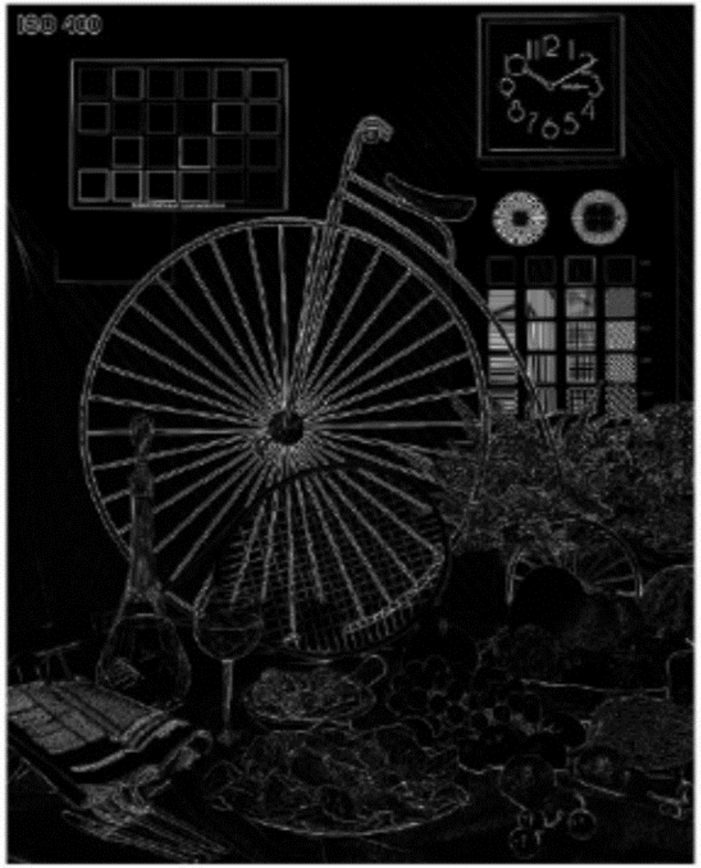
- If I have a one-dimensional signal and calculate the second derivative, I find that at the side it passes through zero.



Laplacian

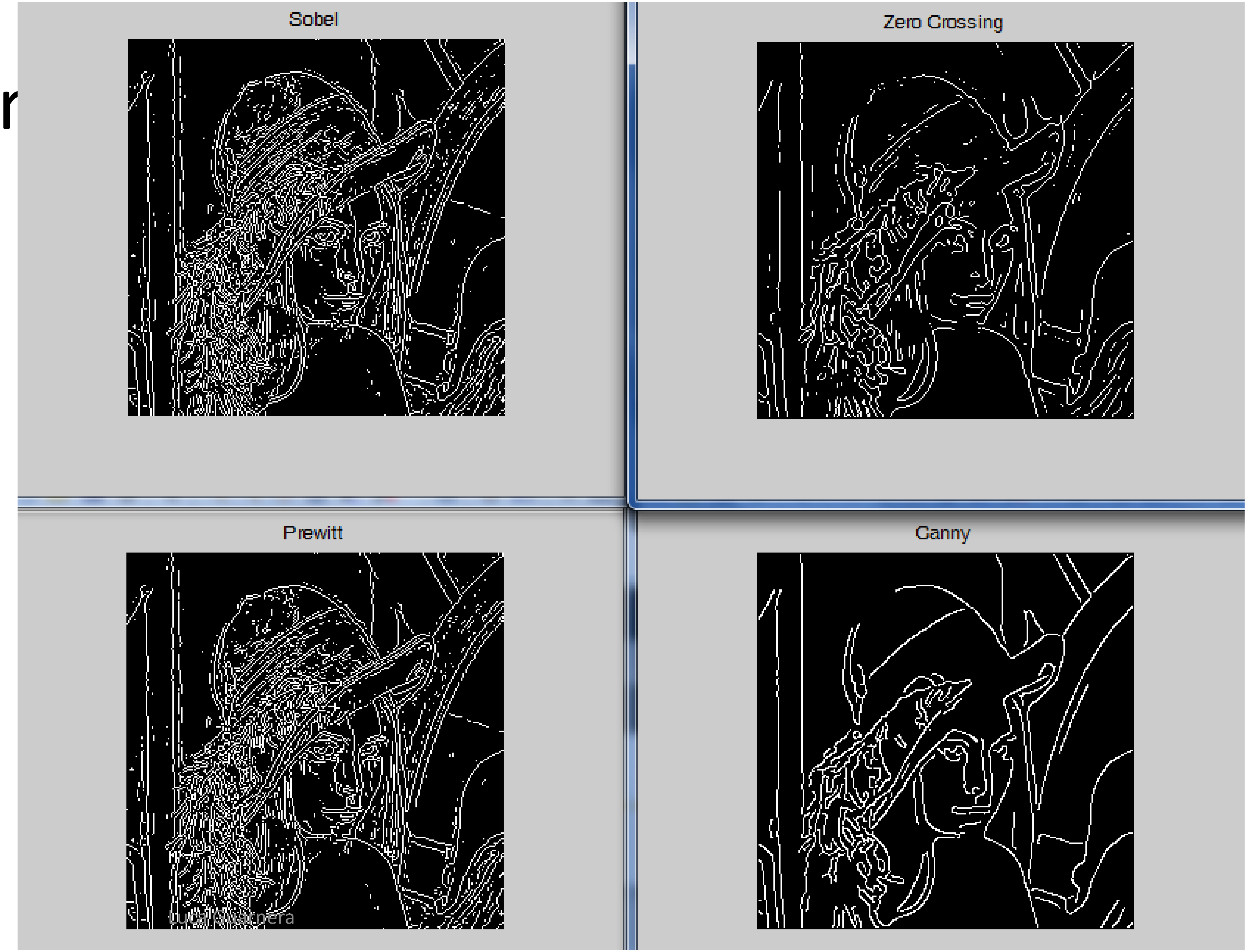
-1	0	-1
0	4	0
-1	0	-1

Filtro Laplaciano



Laplacian

Confr



Morphology Mathematics

Morphology Mathematics

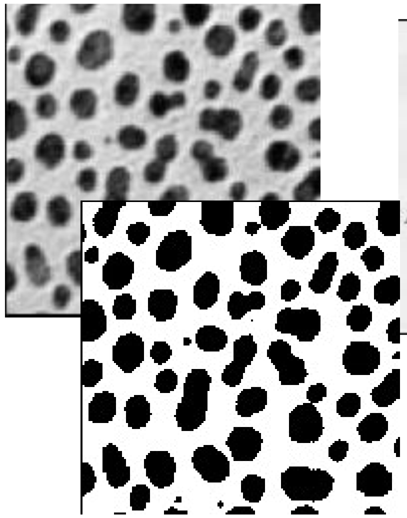
- **Goal**: ***To distinguish relevant shape information from irrelevant information.***

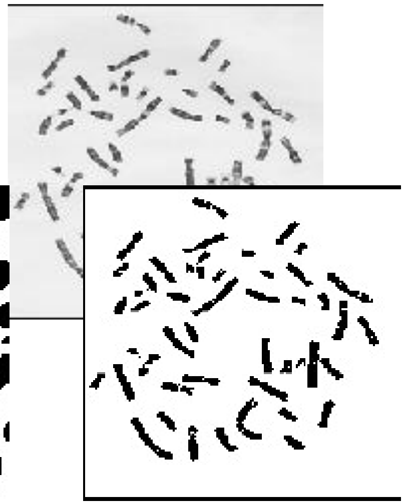
- Most techniques for analyzing and processing the shape of regions are based on making a shape operator that satisfies the required properties.

Examples

The analysis of an image involves the **extraction of measures characteristic** of the image under consideration.

For example, Geometric measures consist of the position of an "object," orientation, area and length of the perimeter...





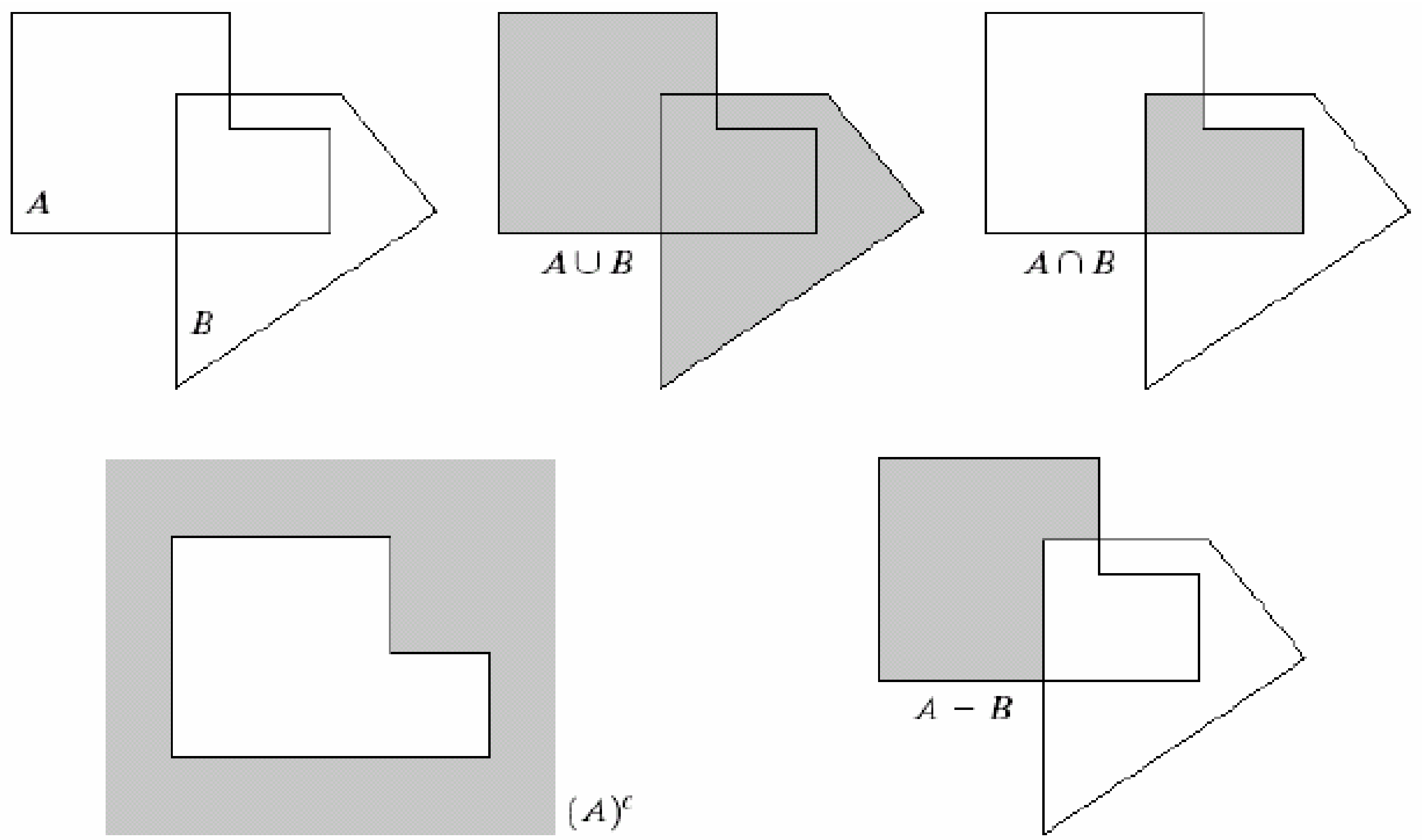
Preliminaries

If an element of A is defined as $a=(a_1,a_2)$ the following expressions are well defined:

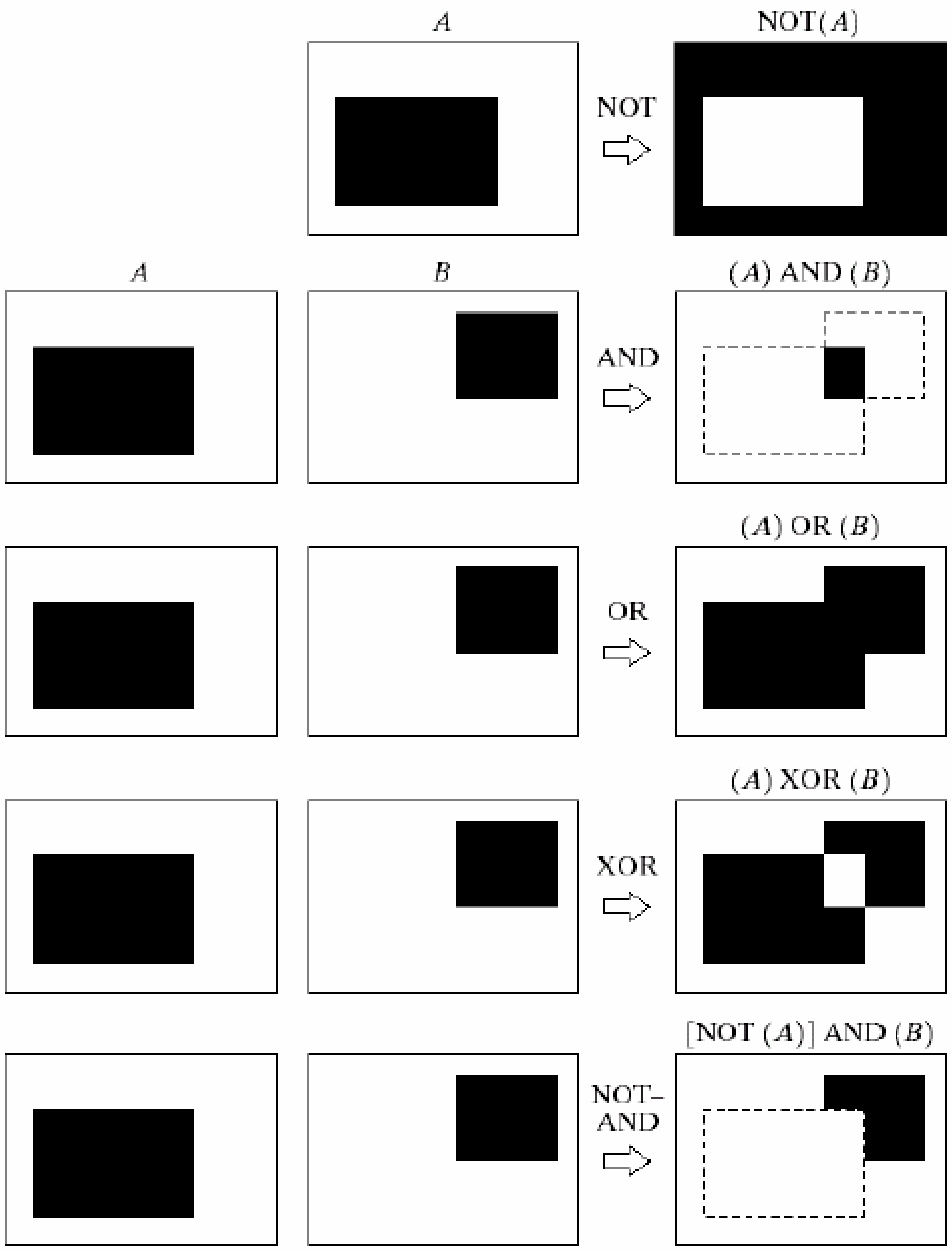
$a \in A$	*a* belongs to the set A;
$a \notin A$	*a* does not belong to the set A;
$A \subseteq B$	A is included in B;
$C = A \cup B$	Union;
$C = A \cap B$	Intersection;
$A \cap B = \emptyset$	Empty intersection;
$A^c = \{w \mid w \notin A\}$	Complementary of A;
$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Insiemistic difference;

Università di Catania

Examples



Logical operations

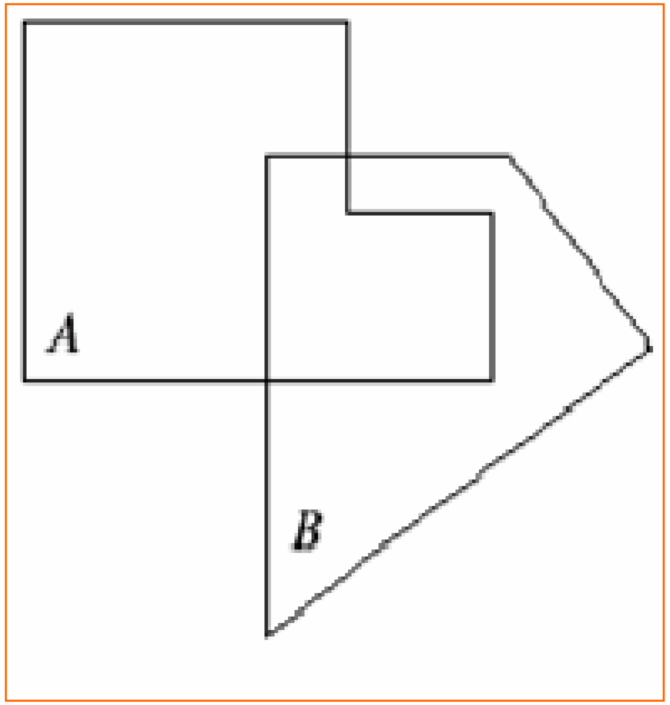


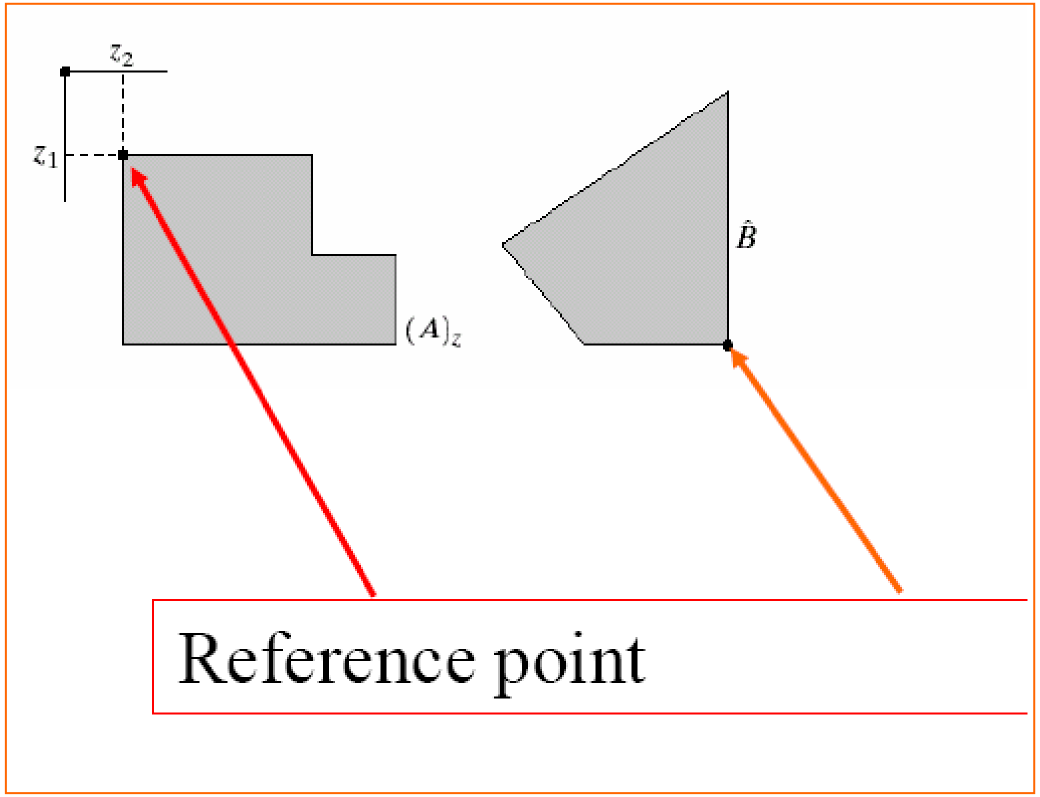
Reflection and Translation

Let A and B be sets in Z^2

$$\hat{B} = \{w \mid w = -b, \forall\, b \in B\},\ Riflessione\ dell'insieme\ B$$

$$(A)_z = \{w \mid w = a + z, \forall\, a \in A\},\quad Traslazione\ dell'insieme\ A$$





Structuring element

The image structure is "probed" with a user-definable shape set (**structuring element**) usually encoded by a small raster image (3×3 or 5×5).

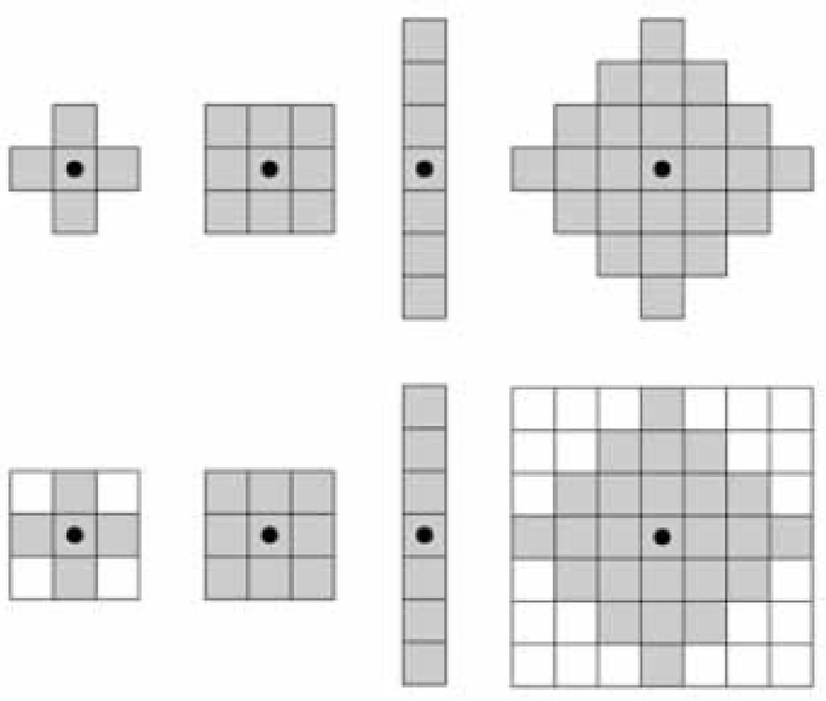
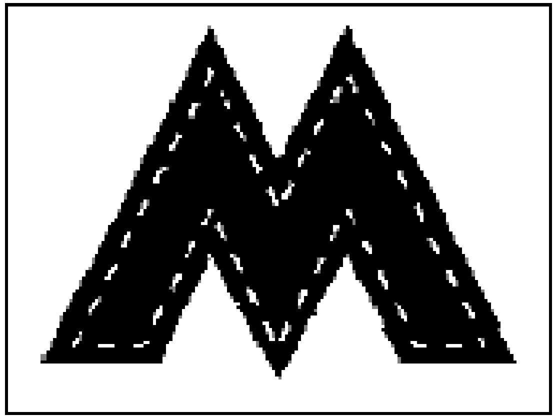


Figura 9.2

Prima riga: esempi di elementi strutturanti. Seconda riga: elementi strutturanti trasformati in matrici rettangolari. I punti indicano i centri degli SE.

Dilatation

Expands objects

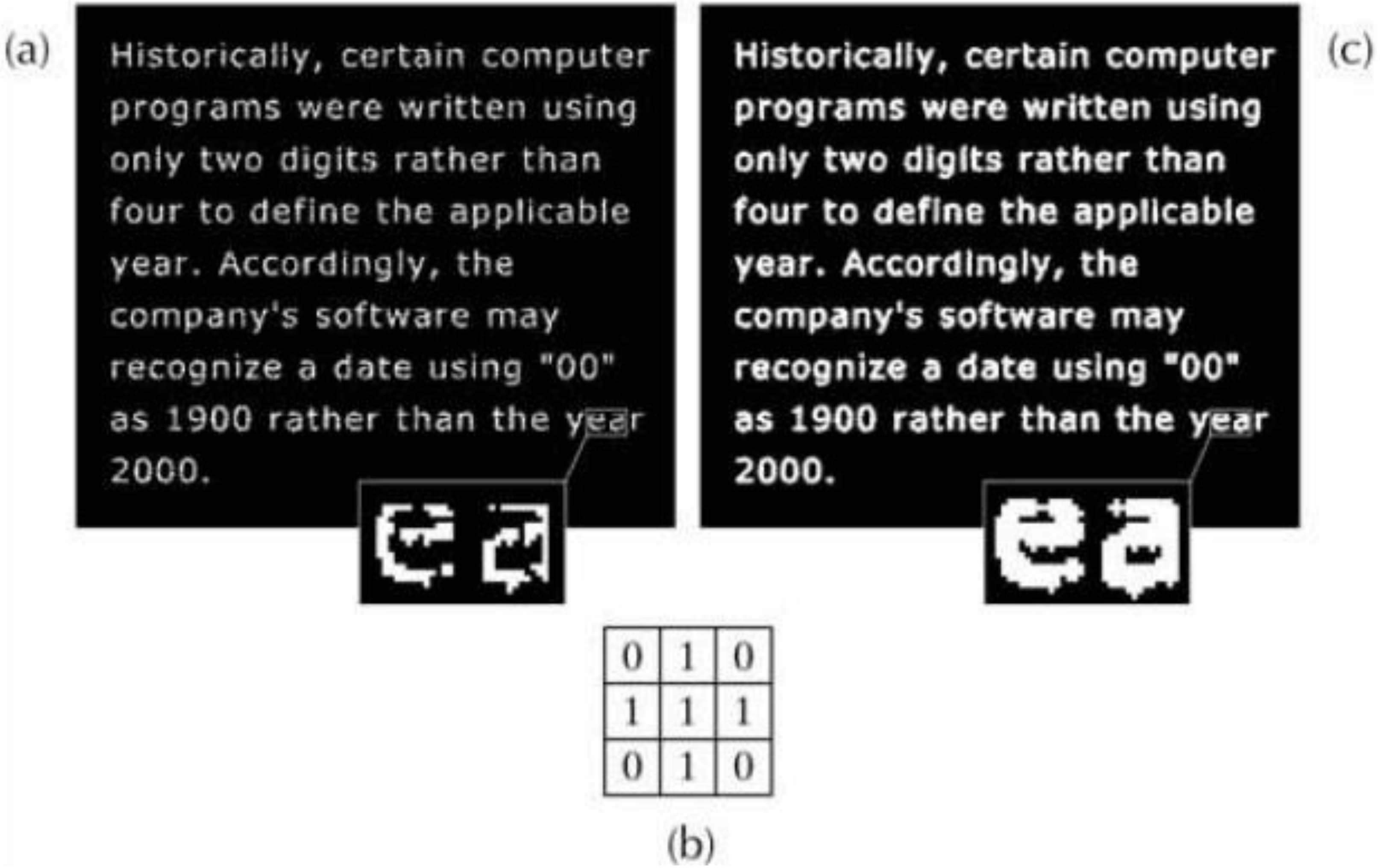


$$A \oplus B = \left\{ z \,\middle|\, \left(\hat{B}\right)_z \cap A \neq \emptyset \right\} = \left\{ z \,\middle|\, \left(\hat{B}\right)_z \cap A \subseteq A \right\}$$

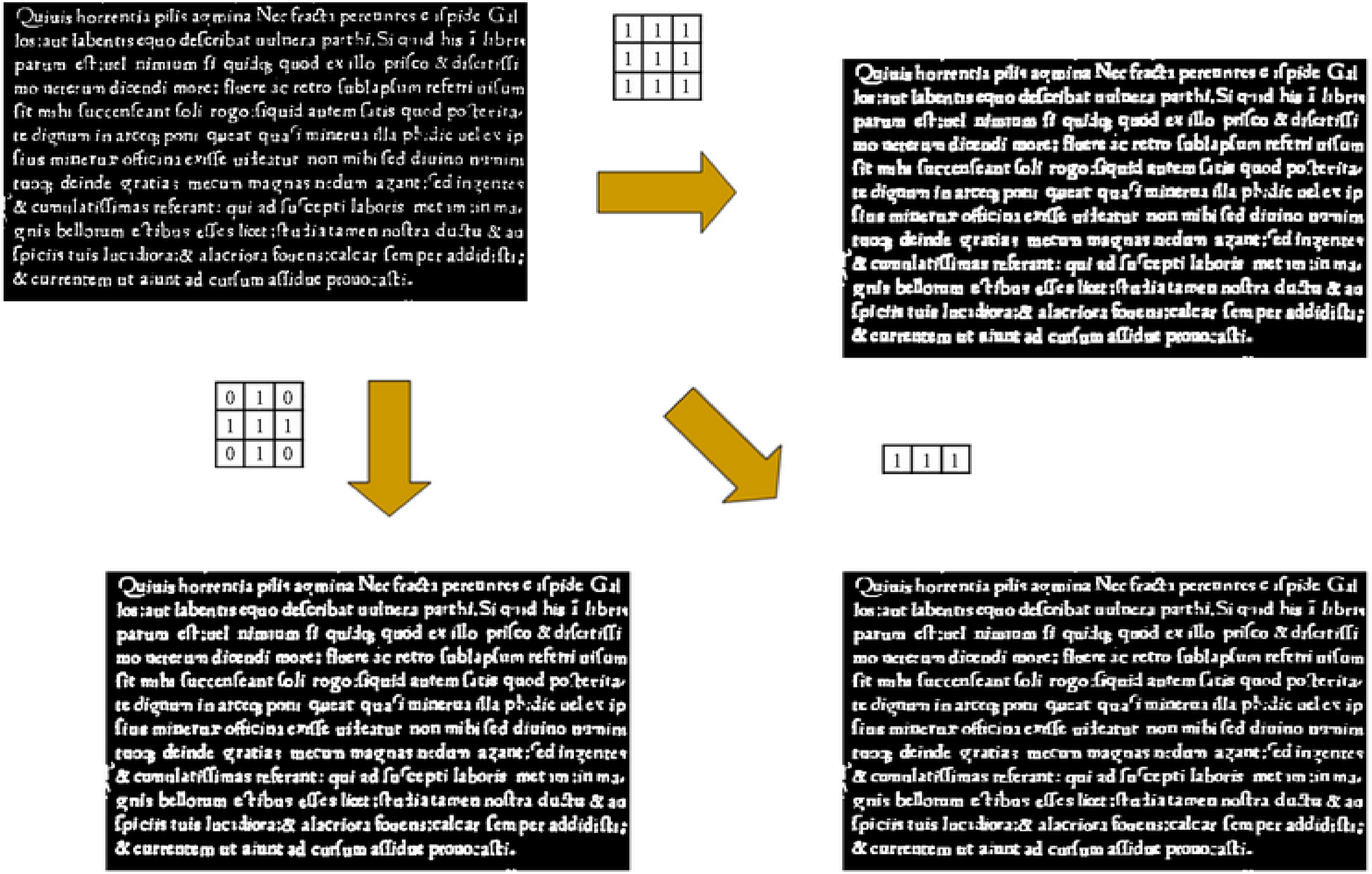
The expansion effect is due to the application of the **structuring element B near the edges**.

It follows from the definition that the structuring element is flipped with respect to its origin, through the reflection operation, and shifted by z positions through a translation. The result of the operator is the set of z positions such that (B^)z intersects at least one element of A.

Example of Dilation

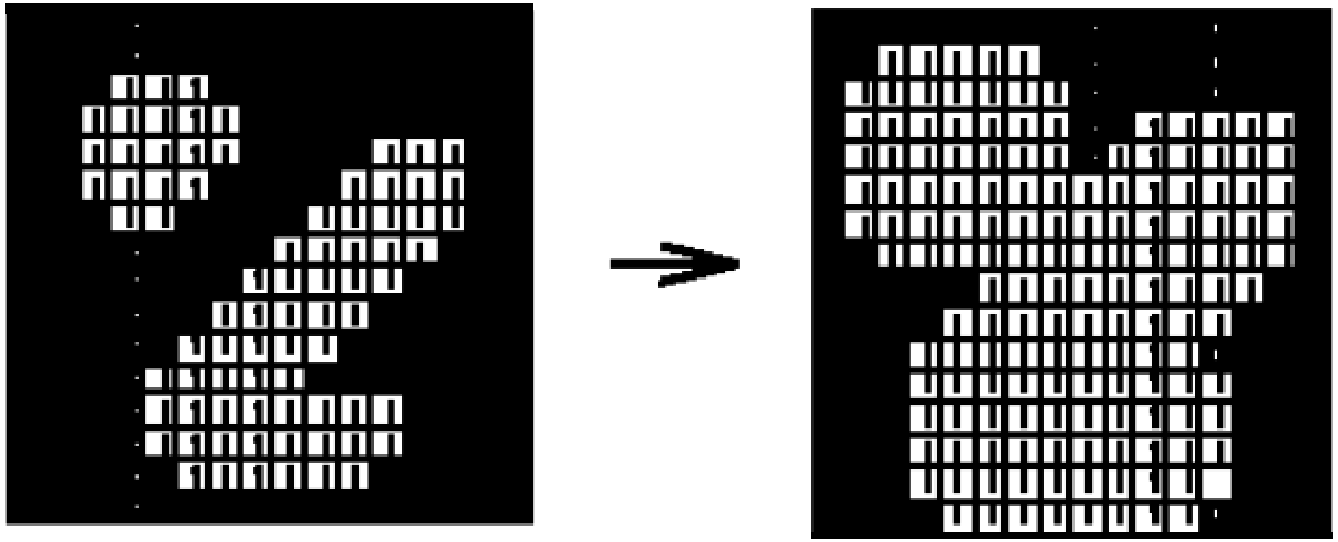


Applications: Filling



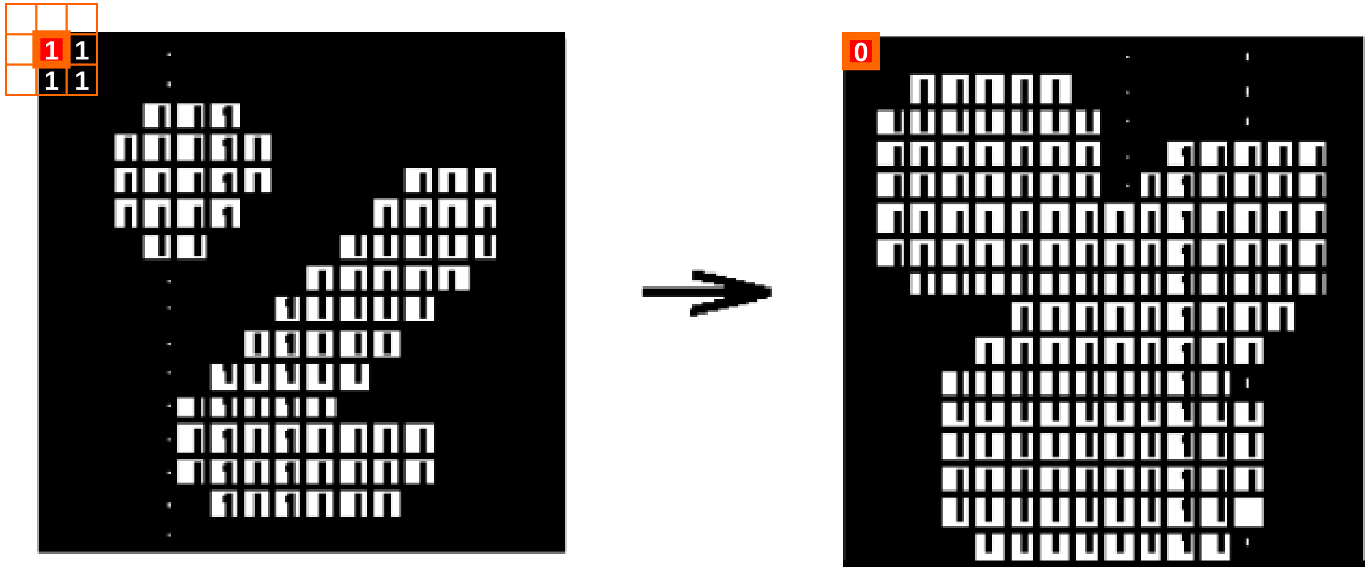
Example of Dilation

> B=ones(3,3)



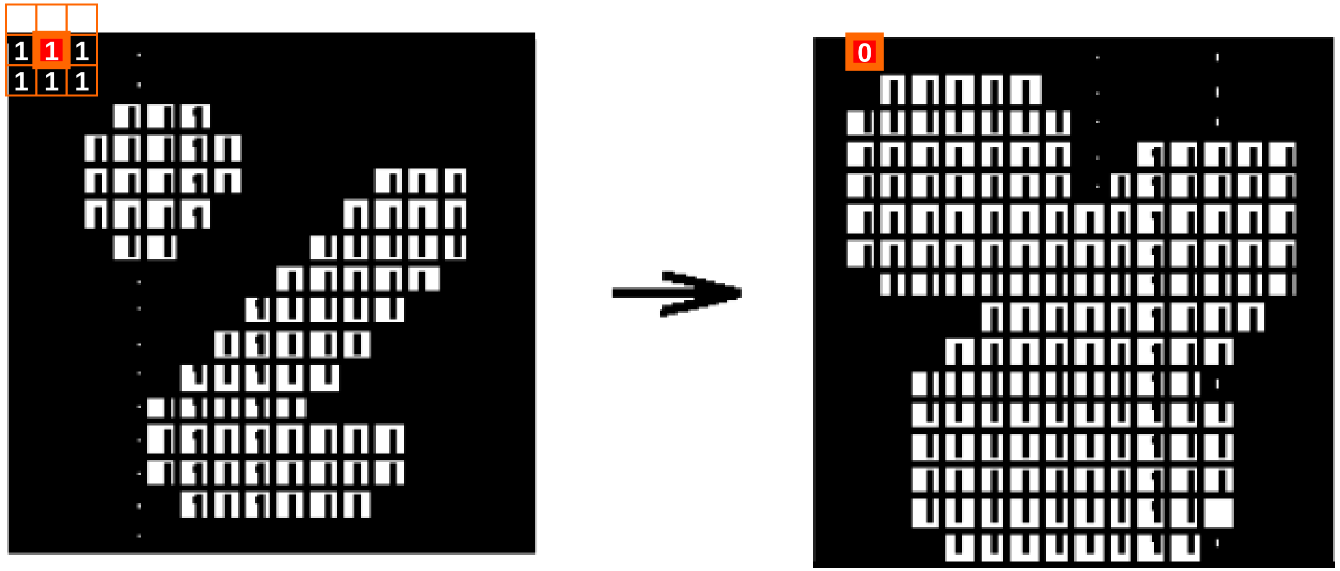
Example of Dilation

> B=ones(3,3)



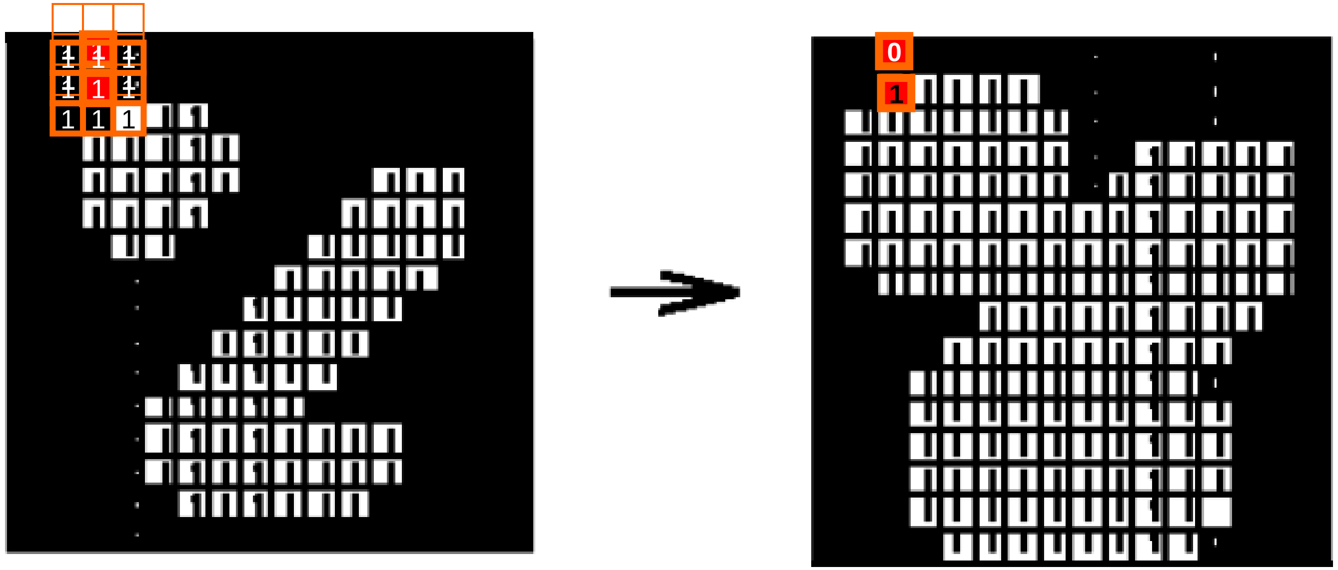
Example of Dilation

> B=ones(3,3)



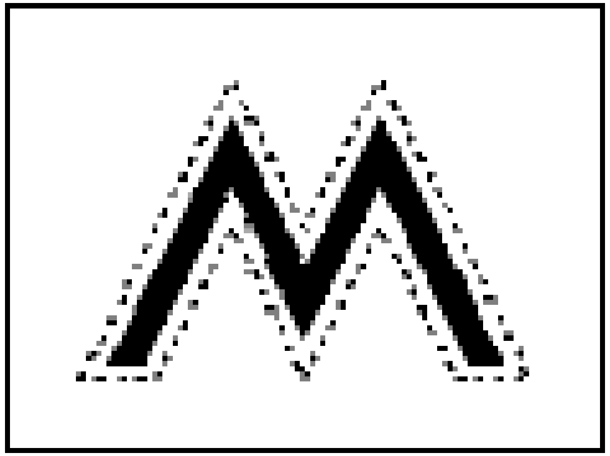
Example of Dilation

> B=ones(3,3)



Erosion

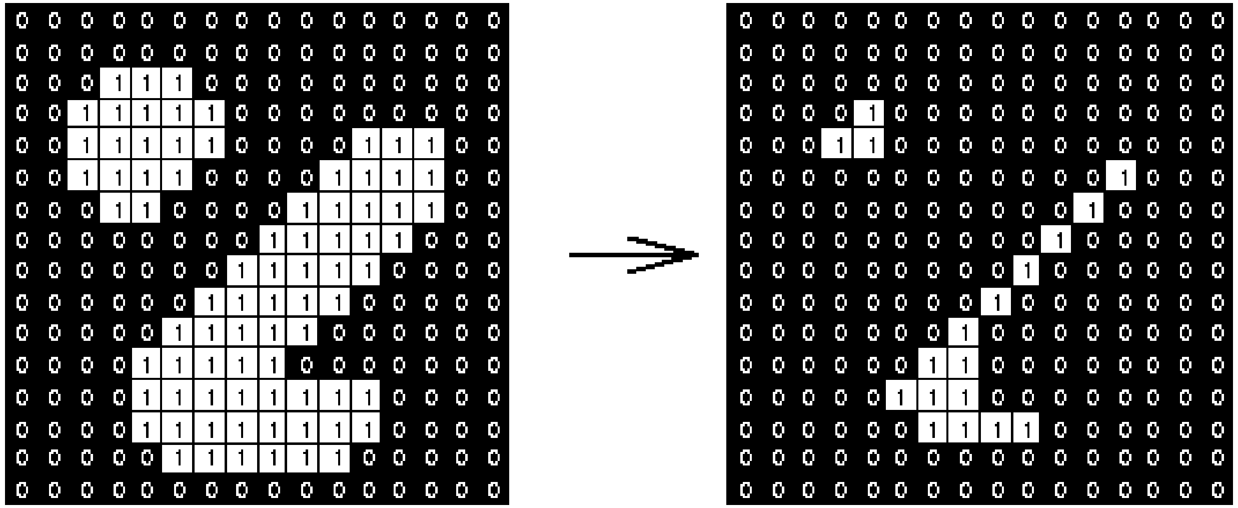
Erodes/shrinks objects



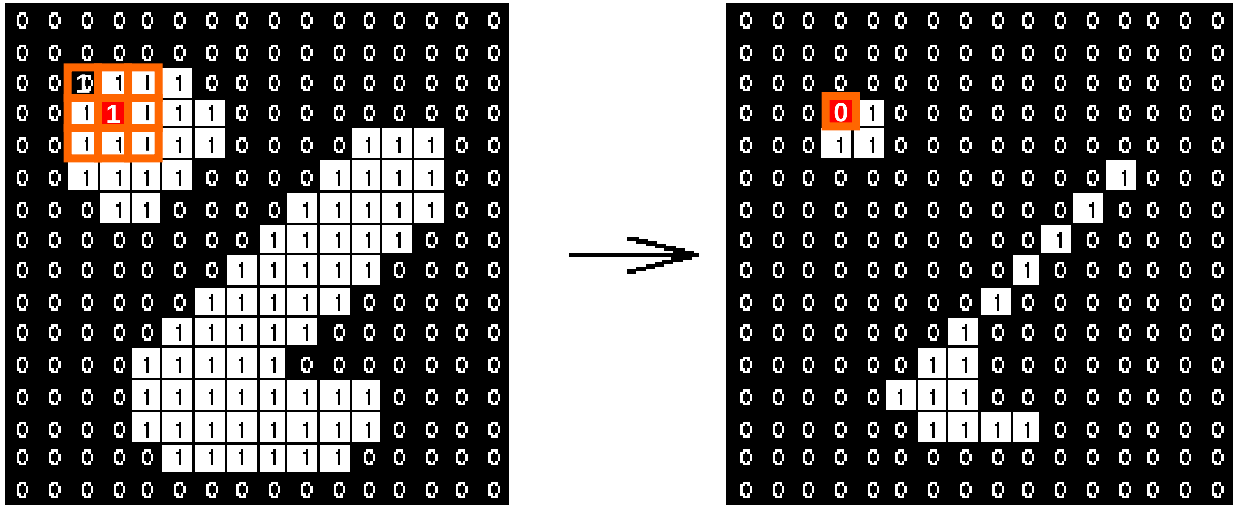
$$A \Theta B = \left\{ z \middle| (B)_z \subseteq A \right\}$$

The erosion effect is due to the fact that when the structuring element B is translated near the edges **it is not completely contained in A**.

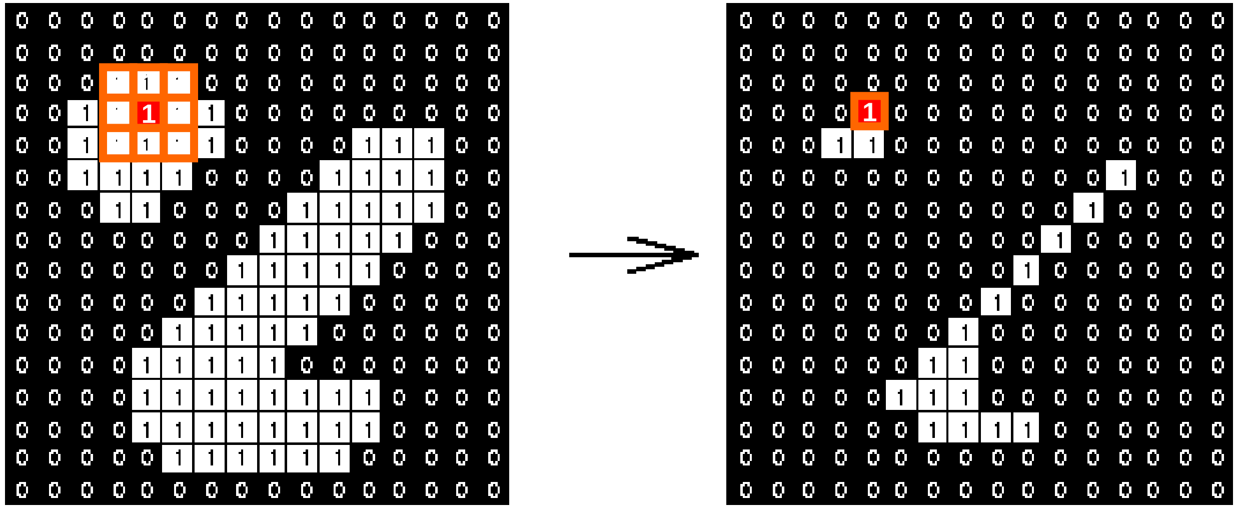
Example of Erosion



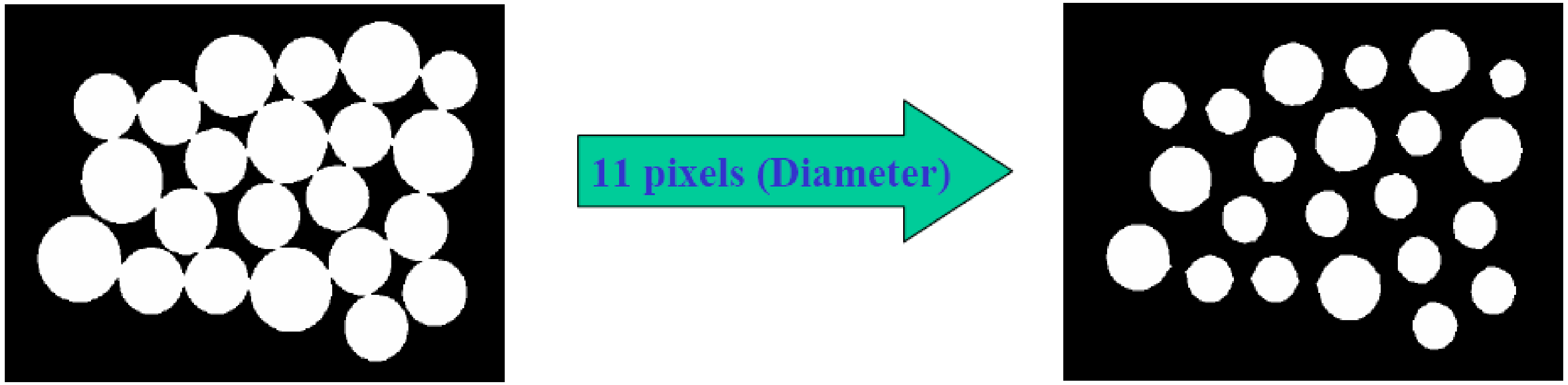
Example of Erosion



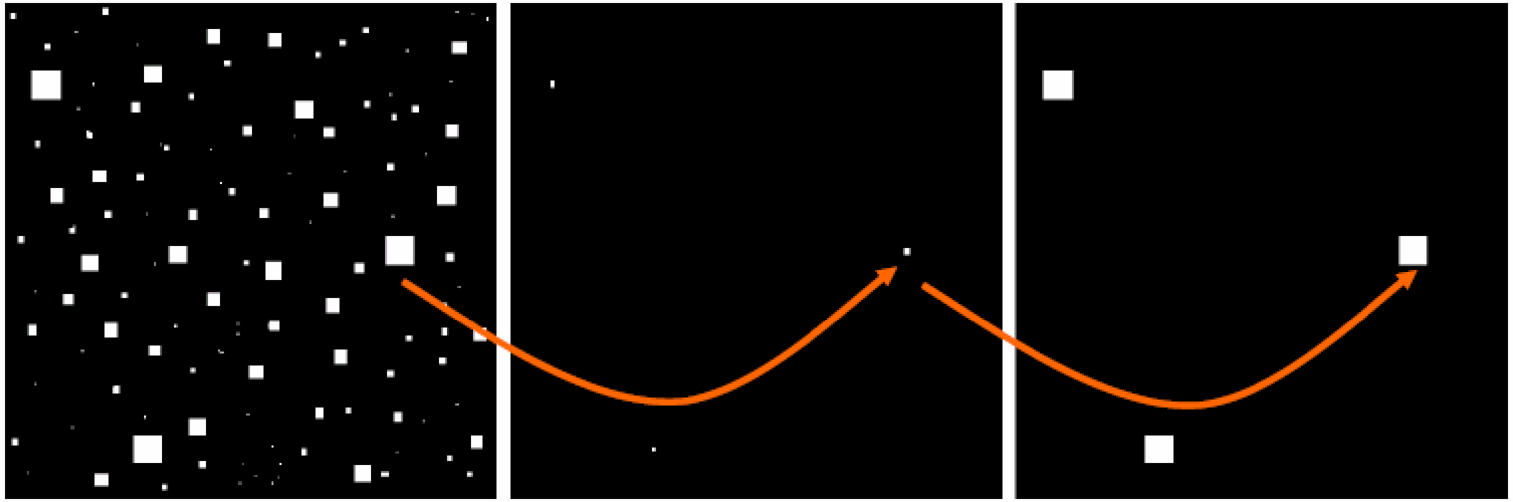
Example of Erosion



Example of Erosion



Particle Size Analysis



Shown on the left is an image containing white squares of size 1,3,5,7,9 and 15. In the center is the output of an erosion process with a structural element of side 13. Then applying an expansion with the same structural element results in an elegant **removal of the initial details**

Estrazione dei contorni

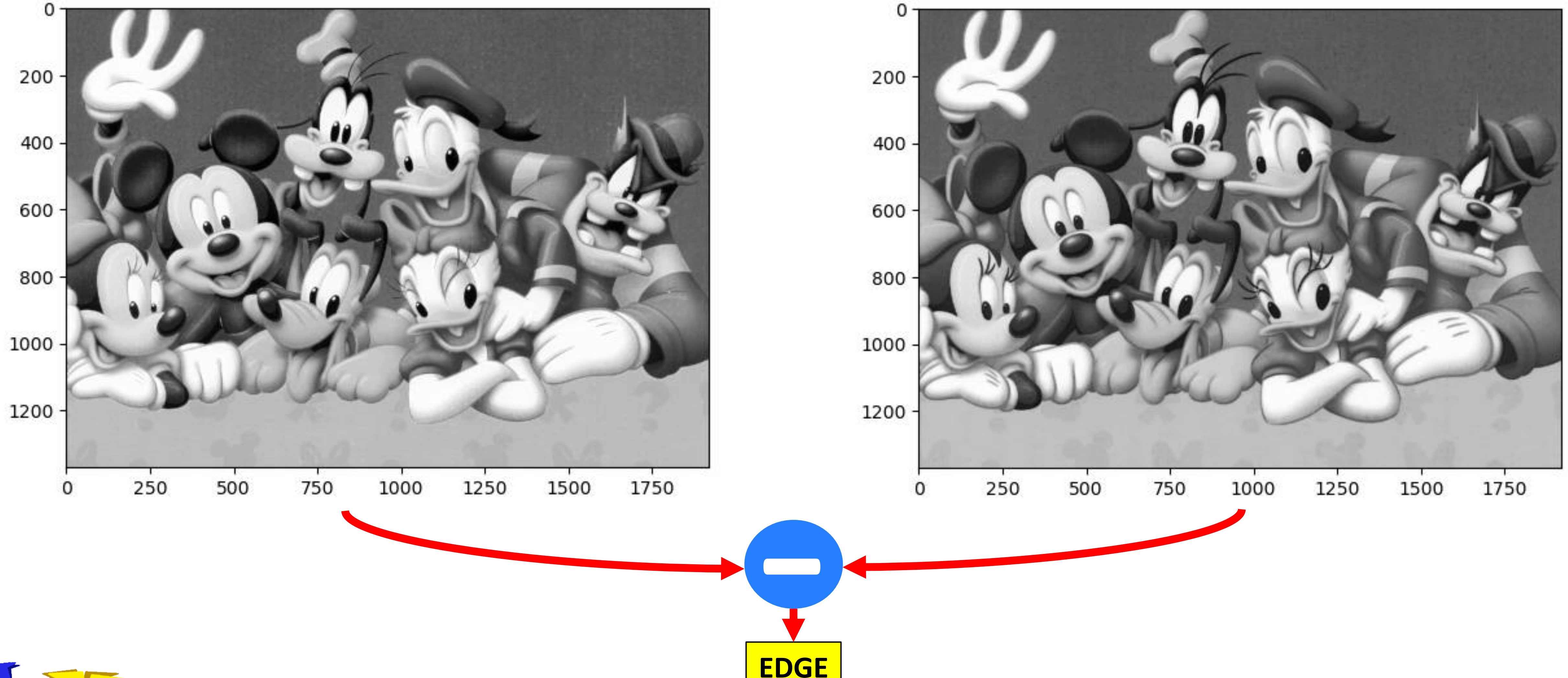
As dilation makes regions thicker and erosion thins them, their difference emphasizes the boundaries between regions. The result is an image in which the edges between objects are clearly seen and in which the contribution of homogeneous regions is not present.

$$g = (f \oplus b) - (f \ominus b)$$

- Binary images
- Grayscale images

Python and OpenCV

Dilate and Erode

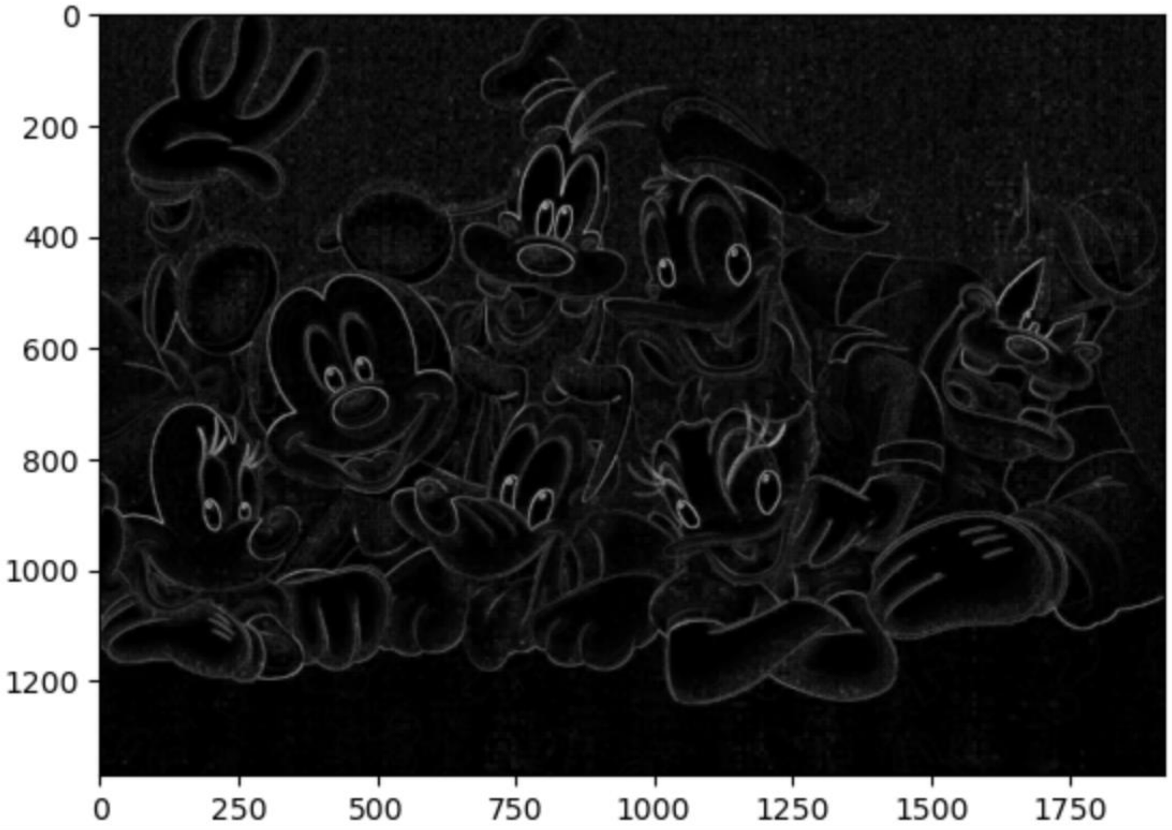


Python and OpenCV

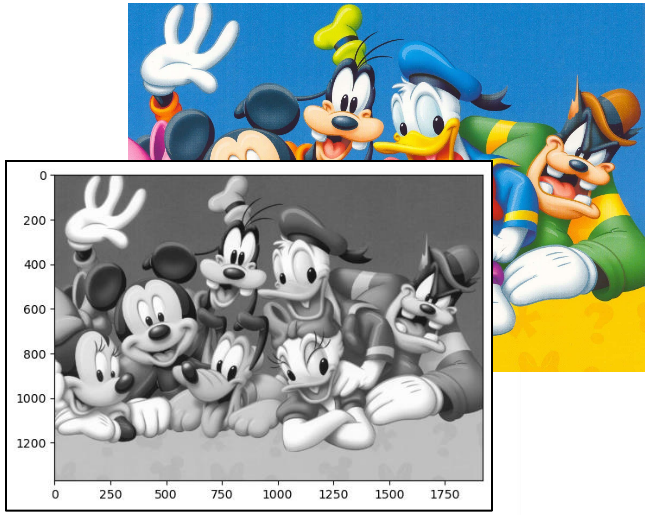
Example with **Gray scale** image

```
edge = dilation - erosion
plt.imshow(edge, cmap="gray")
```

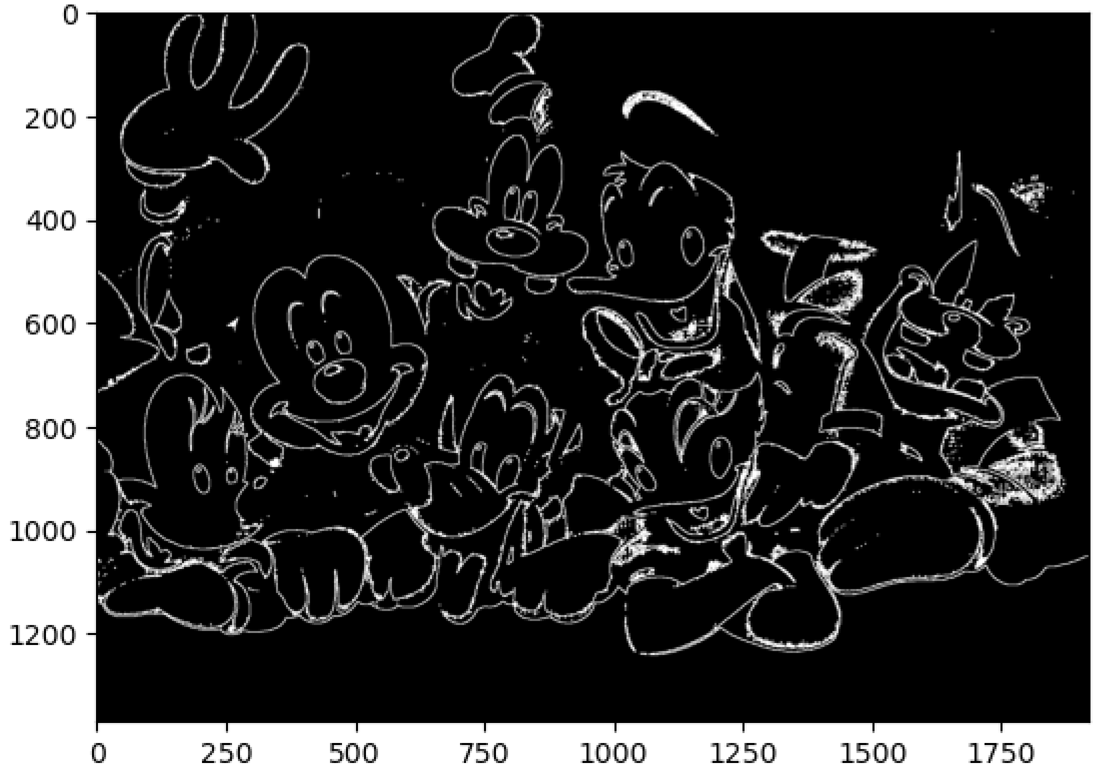
<matplotlib.image.AxesImage at 0x2ca4e0bc190>

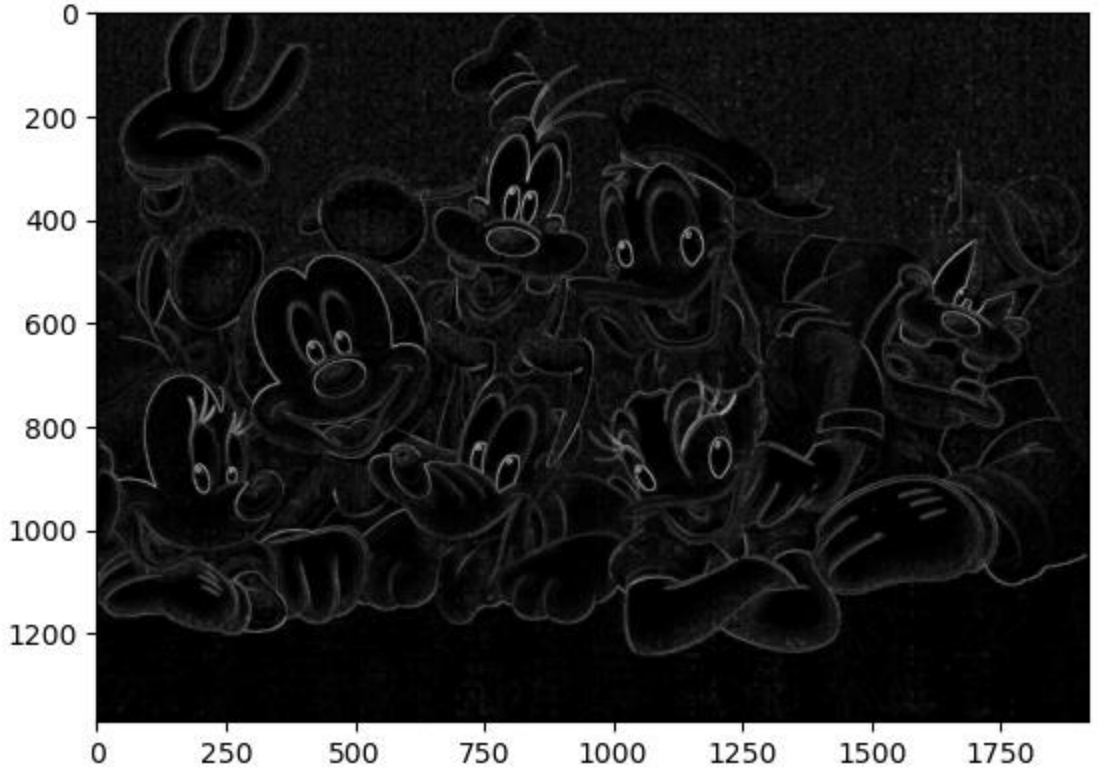


Input



B&W Vs Gray scale





Università di Catania

Fourier and the Frequency Domain

Luca Guarnera, Ph.D.

Research Fellow

luca.guarnera@unict.it

University of Catania

Dipartimento di Economia e Impresa

Introduction

A **periodic function** can be expressed as a **sum** of **sines** and/or **cosines** of different frequencies and amplitudes (Fourier Series).

Even a **nonperiodic function**, (under certain conditions) can be expressed as an **integral** of **sines** and/or **cosines**, multiplied by appropriate weight-functions (Fourier Transform).

Jean Baptiste Joseph Fourier
(Auxerre, 1768 –Paris, 1830)

Università di Catania

Introduction

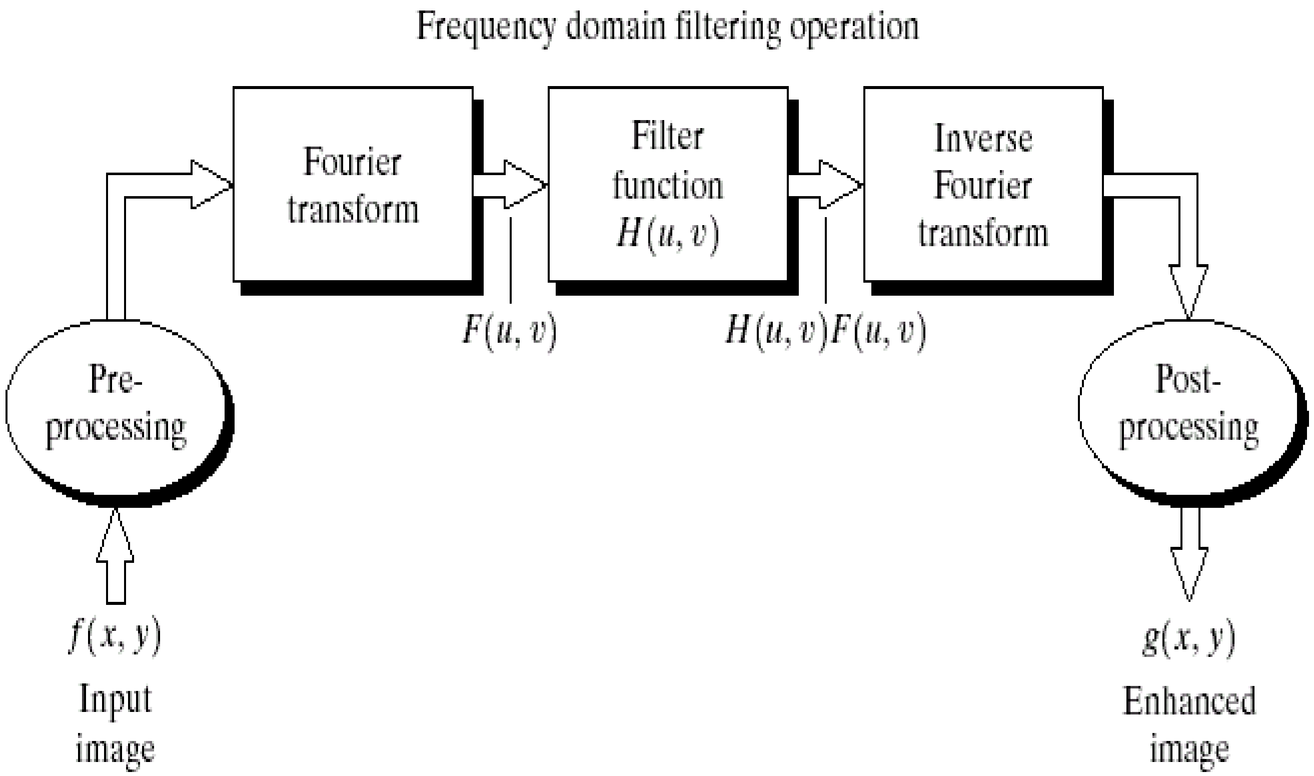
Both the Fourier series and the Fourier Transform share the fact that a **function can be "reconstructed" (recovered) by a simple inversion process without loss of information**.

That is, it is possible to work in the so-called Fourier domain and return to the original domain of the function in a completely natural way.

- Originally, Fourier analysis found application in the field of heat diffusion where it enabled the formulation and solving of differential equations of certain physical phenomena in a completely original way.

Introduzione

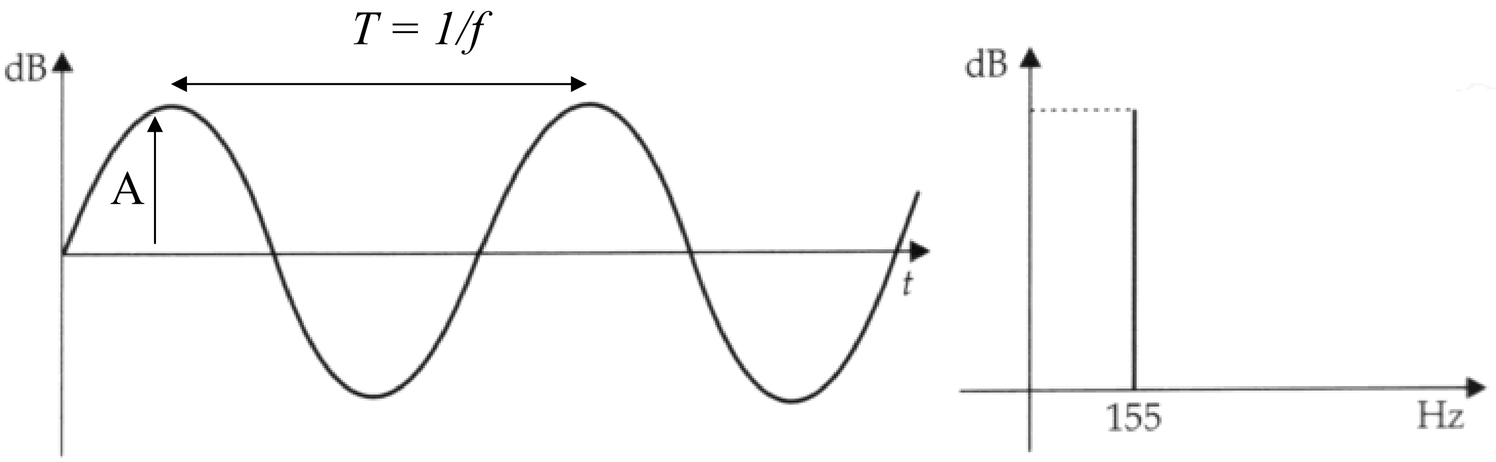
With the emergence in the 1960s of the **FFT** (**Fast Fourier Transform**), the field of **digital signal processing** (**DSP**) underwent a **real revolution**, and today these concepts find application in a wide variety of industrial fields, **from medicine to telecommunications**, etc.



Images and Signals

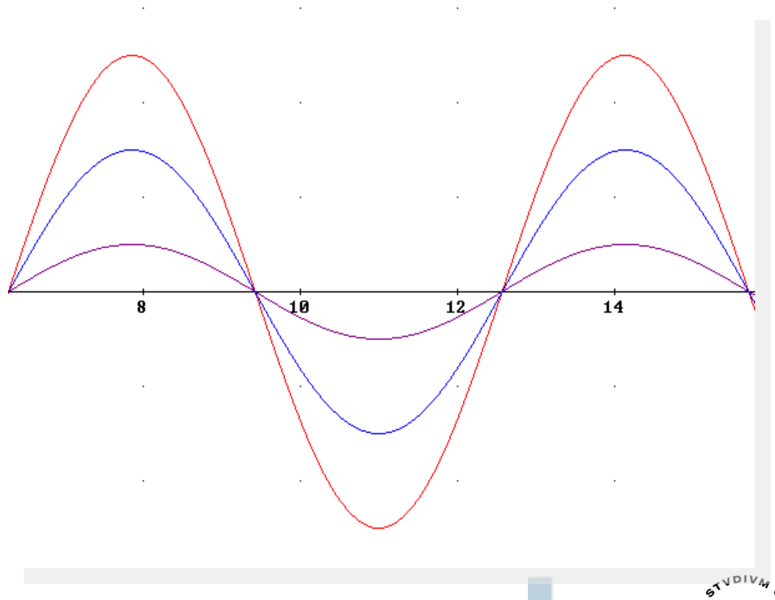
- An image can be viewed as a discrete function in two dimensions whose values represent the gray level of a given pixel.

- The "image" function can be viewed as a signal, that is, a variable function in a domain with its own frequency (constant or variable).



Time Domain

Frequency Domain



- **Amplitude (A) expressed in decibels dB;**
- **Period (T) expressed in seconds;**
- **Frequency (f) number of cycles (waves) per second; measured in Hertz Hz**

Discrete Fourier Transform

In the 2-D case the **transformed / anti-transformed** pair of the two-dimensional sequence f(x,y) takes the following form:

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad per\ \ u = 0,1,...,M-1\ v = 0,1,...,N-1$$

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad per\ x = 0,1,...,M-1\ \ y = 0,1,...N-1$$

u and v are the indices related to the discretized frequency axes, while M and N are the dimensions (in pixels) of the image.

Euler formula

- For each real number **x** we have:

$$e^{ix} = \cos x + i \text{ sen } x$$

- And so:

$$e^{-ix} = \cos x - i \text{ sen } x$$

Trasformata di Fourier

Since the F transform has complex values, it can be expressed in terms of its real part and its imaginary part.

Transform Spectrum

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

Phase Angle

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

Spectral Power

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

Università di Catania

Dynamic range

- When **viewing the Fourier spectrum** as an **intensity image**, it typically manifests much greater dynamics than can be reproduced on a typical display, so only the brightest parts of the spectrum are visible. For example, the spectrum of Lena's image varies between 0 (approximately) and 6.47x106.

- By performing the normalization necessary to display it with L=256 levels of gray, only very few very bright parts are visible.

- This can be remedied, as is known, by means of a **logarithmic type of compression**, displaying, instead of the spectrum, a function of the type:

$$D(u,v)=c \log(1+ F(u,v))$$

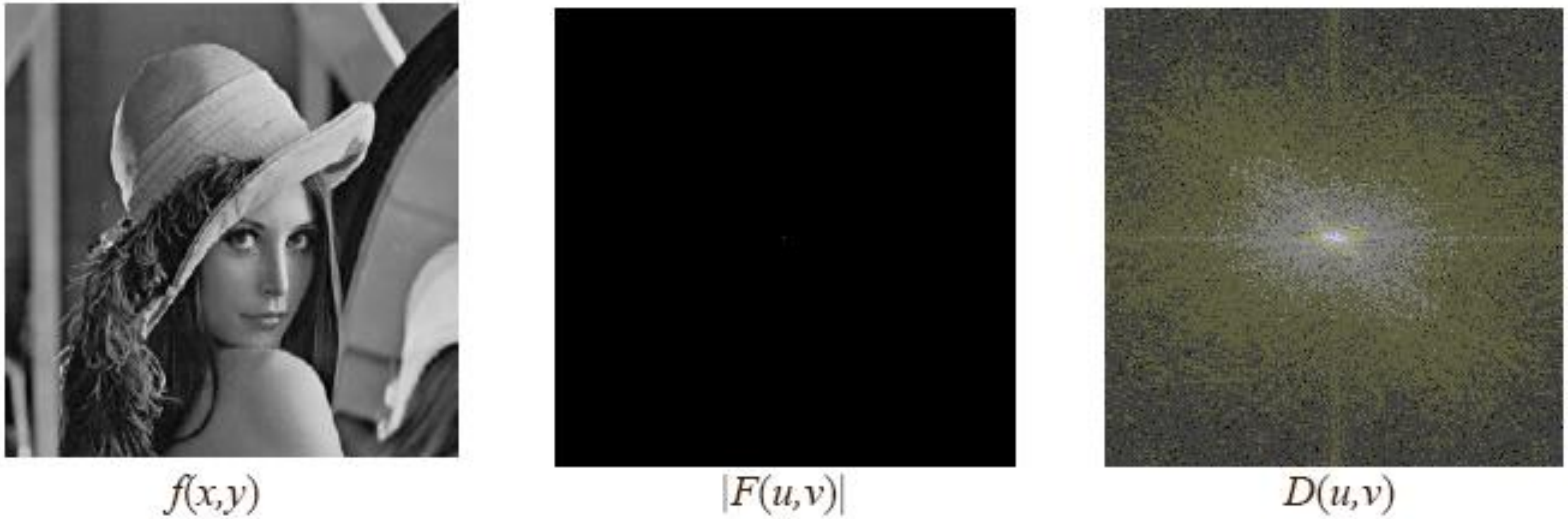
- *c* is a scaling constant, which must be chosen appropriately so that the transformed values fall in the desired range, that is, in [0, L-1]

Università di Catania

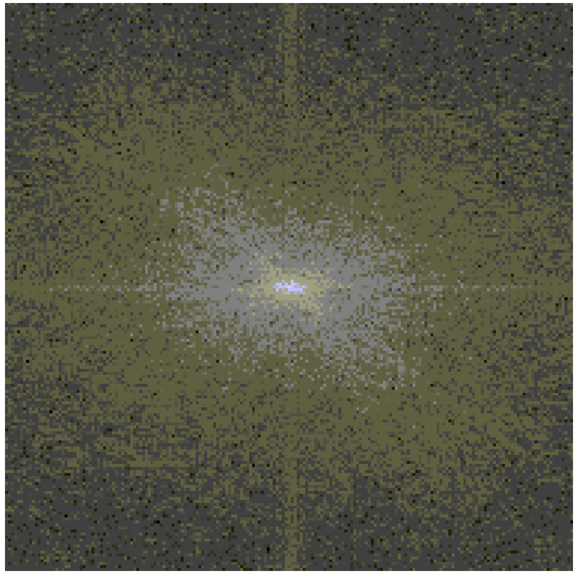
Dynamic range

Since 0 < |F(u,v)|< R = 6.47x106, we have 0 < D(u,v)<clog(1+R). Since R>>1, as, moreover, is normally the case for the Fourier spectrum of an image, one can put c log R=L-1, whence c= (L-1)/log R= 255/log(6.47x106) = 16.26

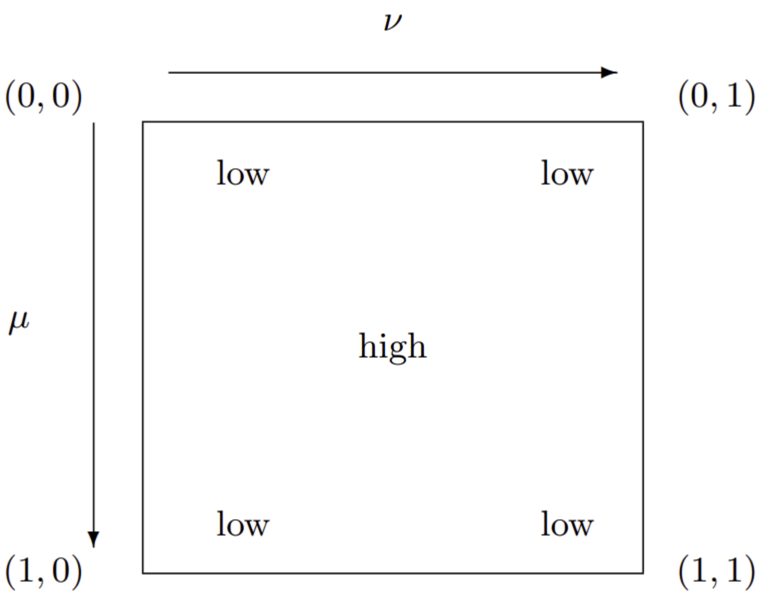
Therefore D(u,v) has all values in the range [0, 255], and this allows for the visualization of much more detail.

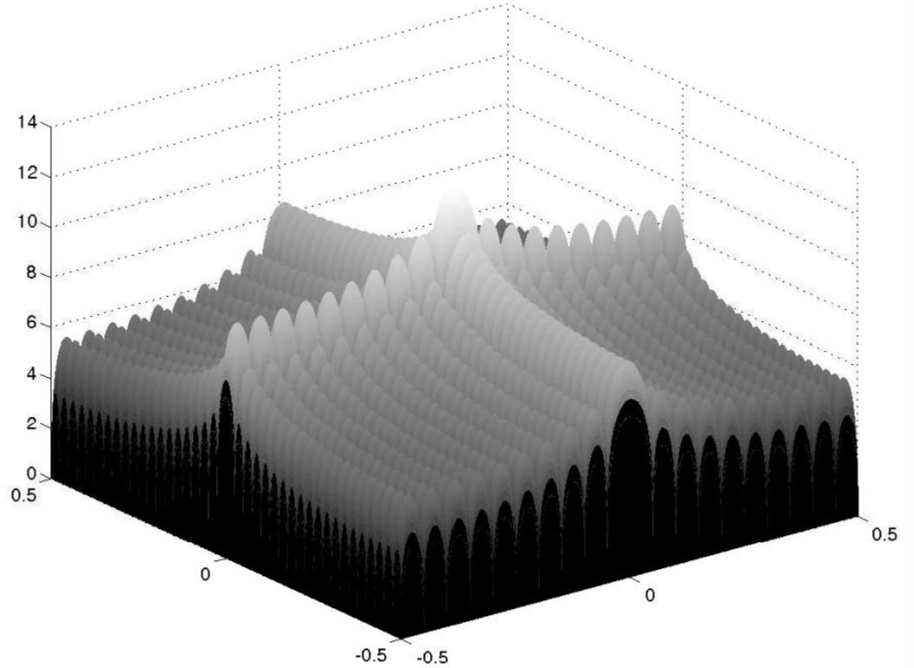


Details

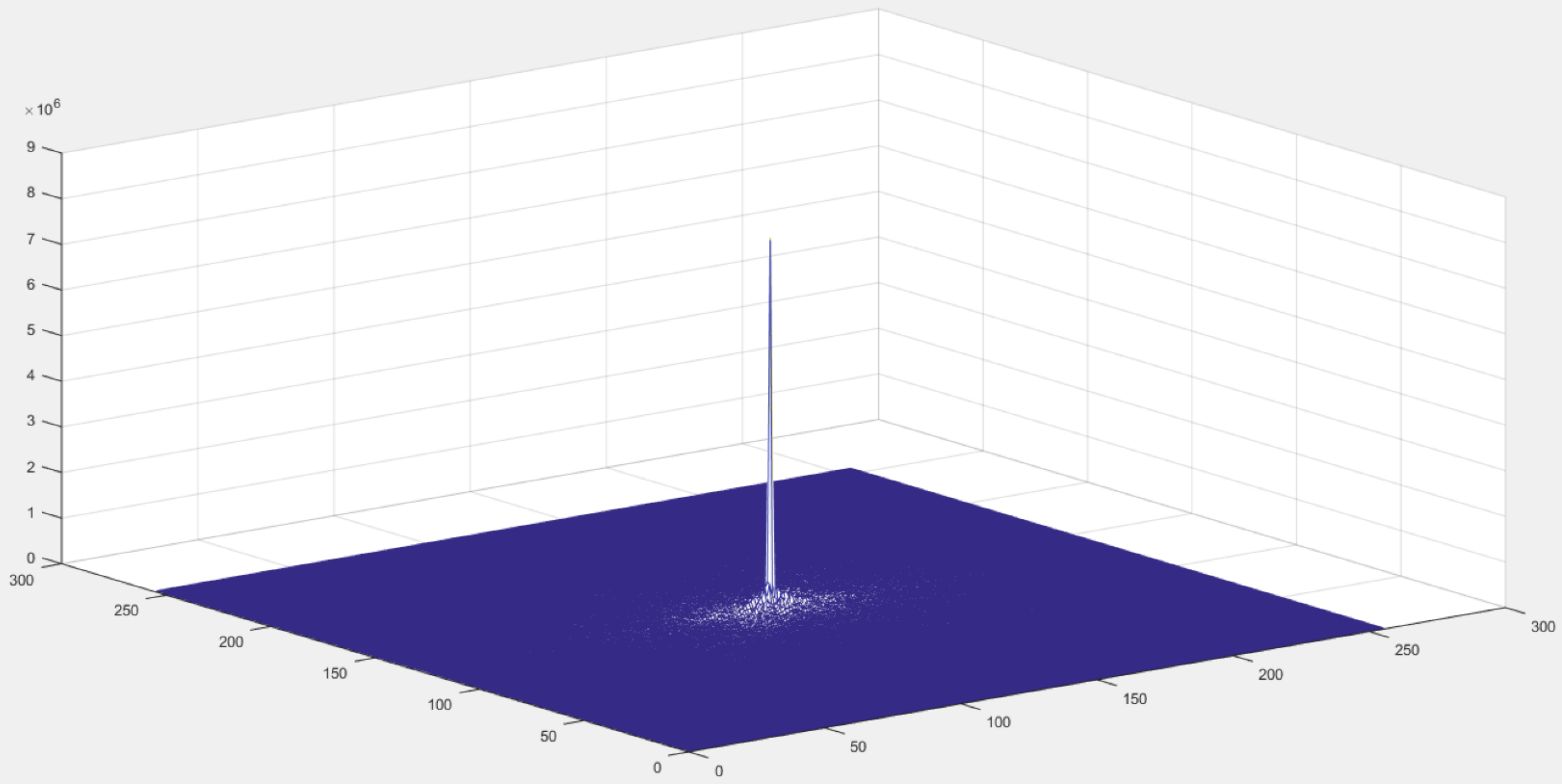


$D(u,v)$

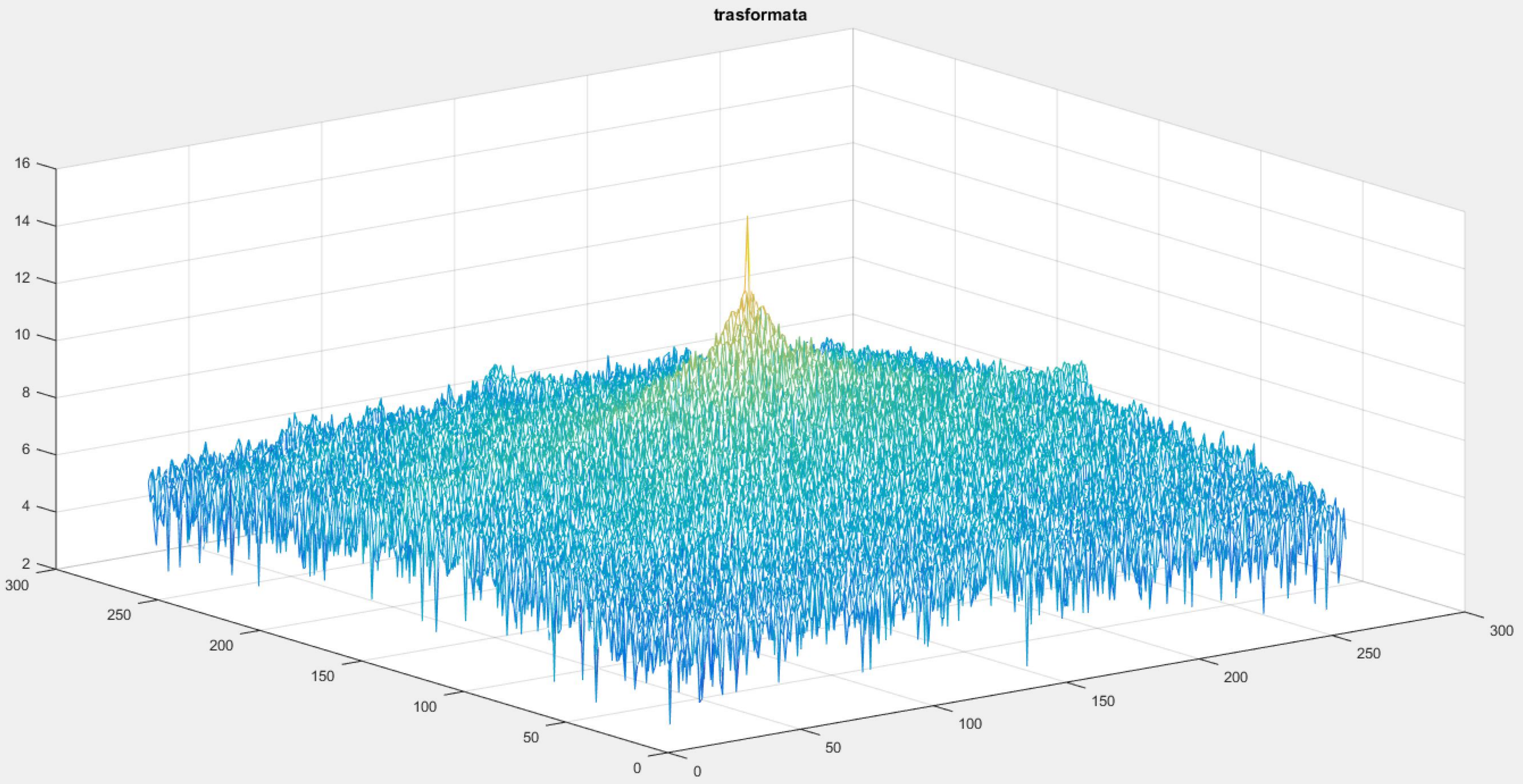




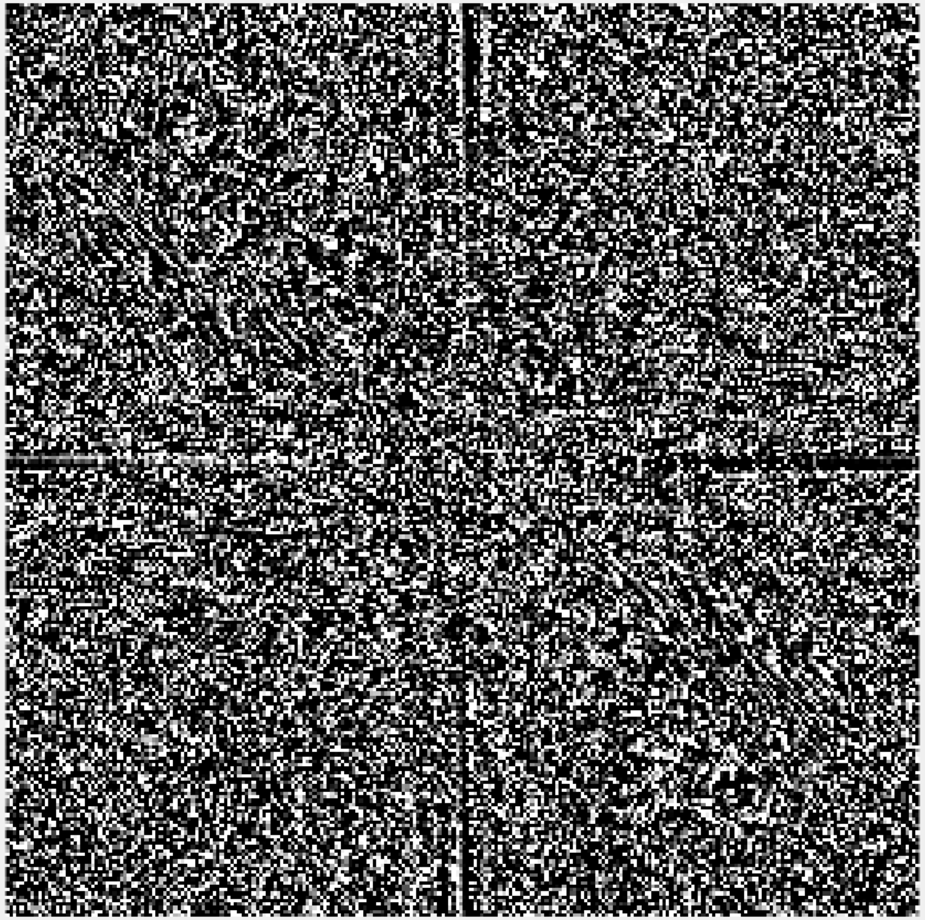
Lena's module



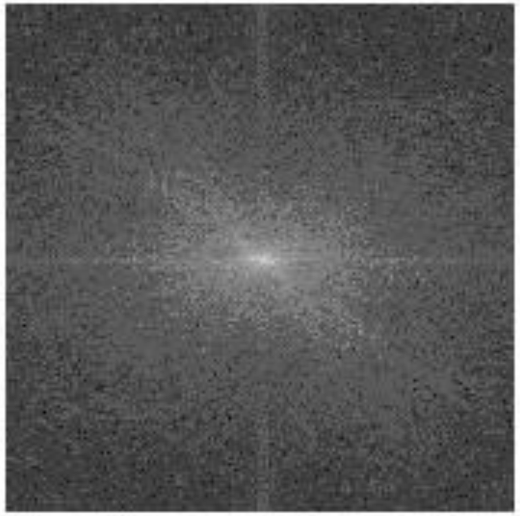
Logarithm of Lena's modulus



Lena Phase



$f(x,y)$



$|F(u,v)|$

$\Phi\,(u,v)$

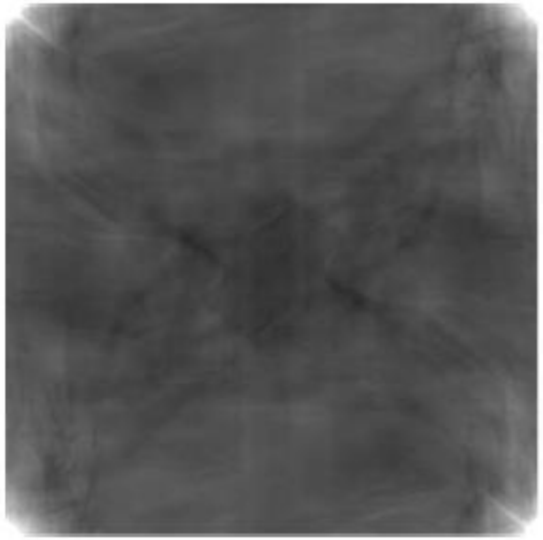
The spectrum display is actually about not |F(u,v)| but a logarithmically compressed version of it. Otherwise only a dot in the center would be seen.

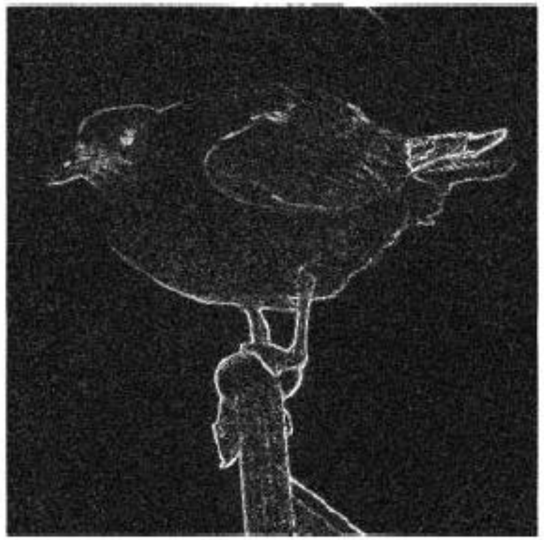
The **amplitude** contains information concerning whether a certain **periodic structure** is present in the image.

The **phase** contains the information regarding where the **highlighted periodic structures in the DFT are located**. So it is much more significant than it may appear in the image.

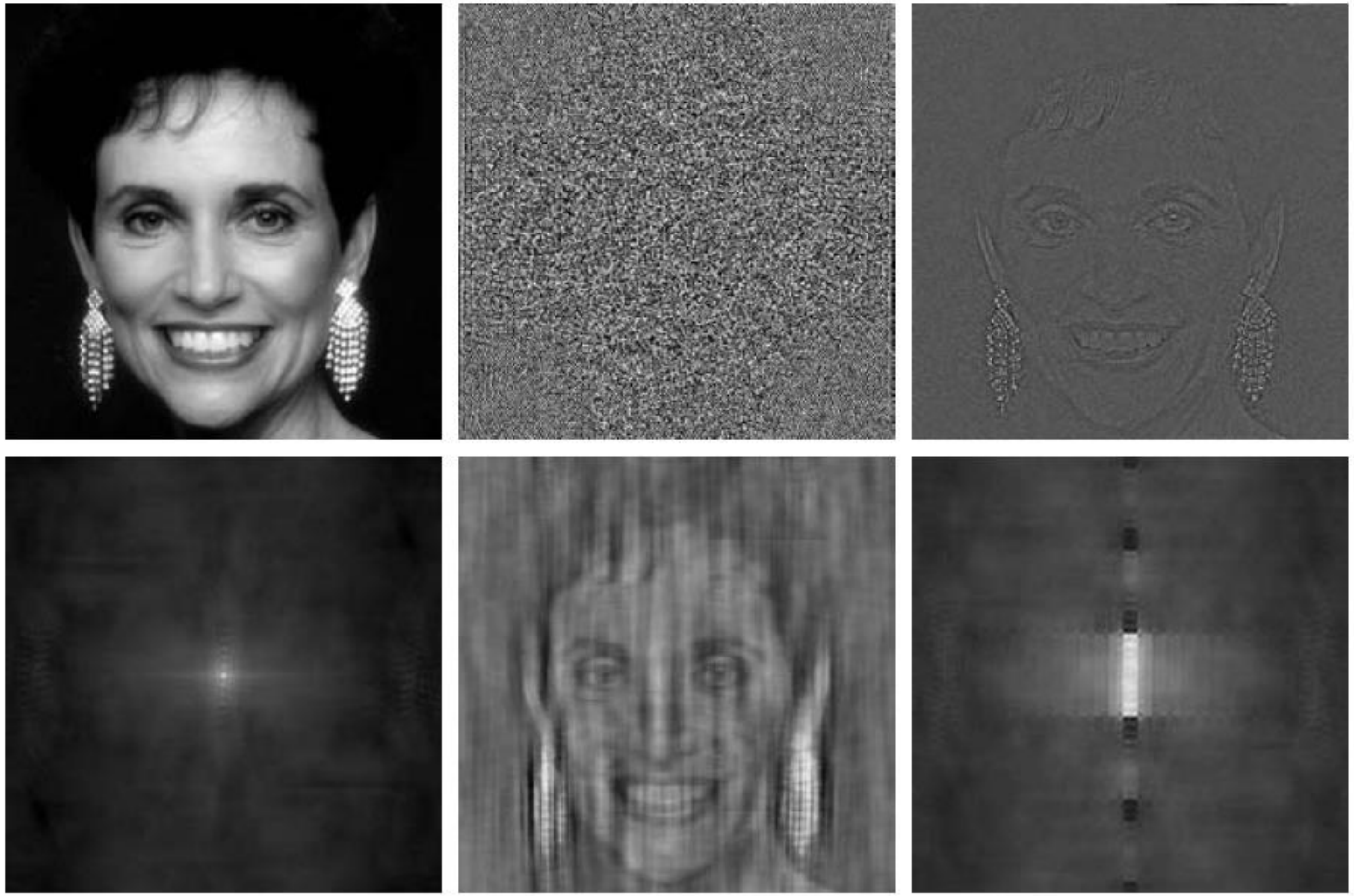
Amplitude vs. phase

Reconstruction alone module





Single-stage reconstruction

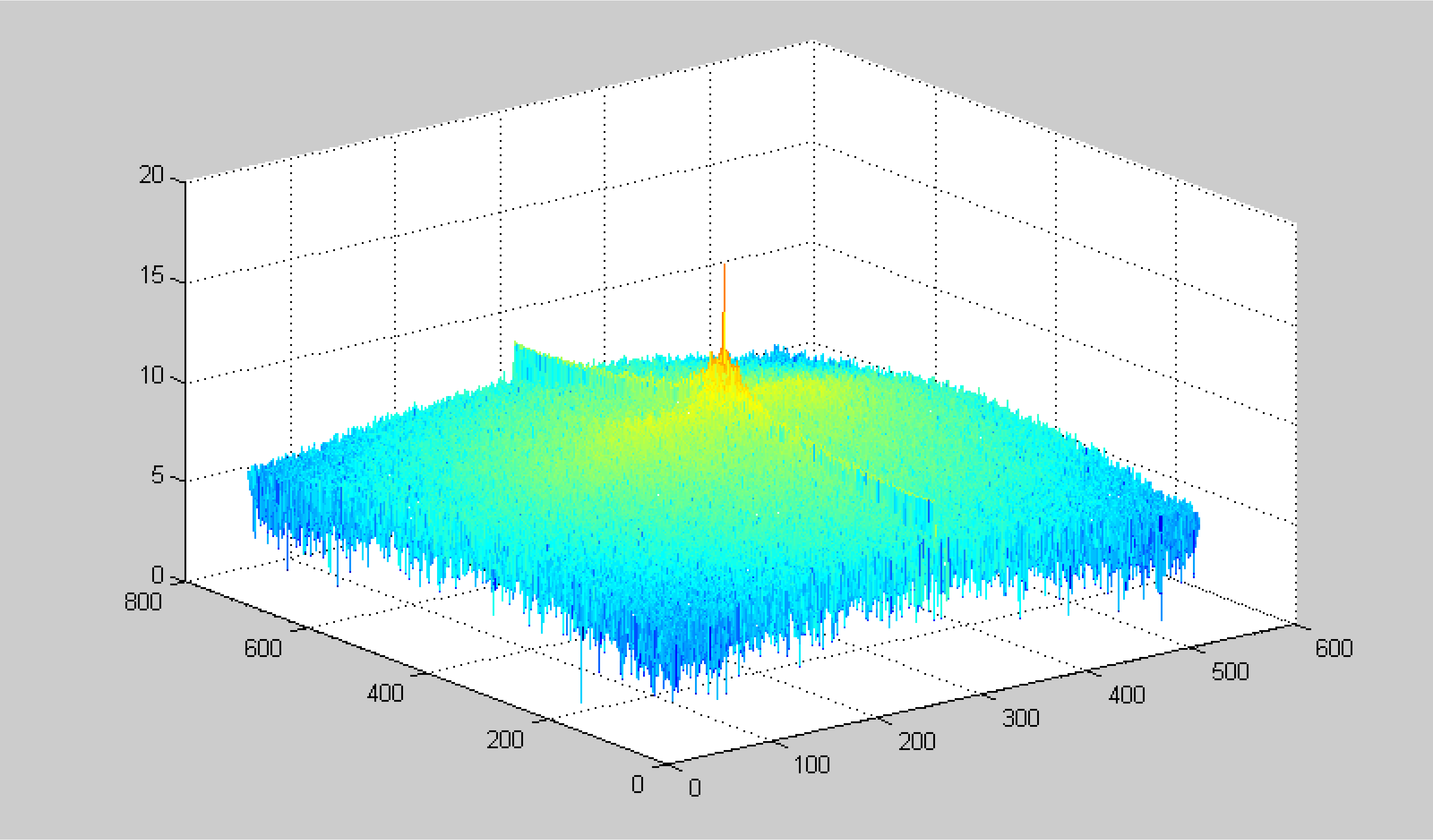


a	b	c
d	e	f

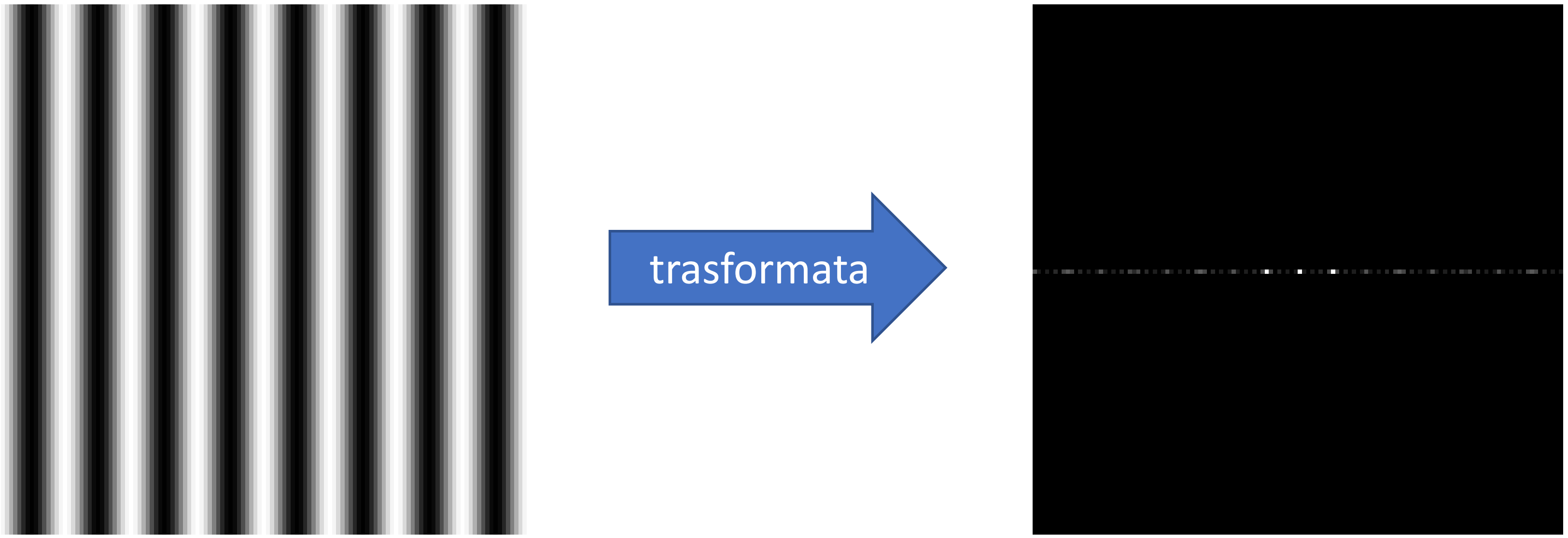
FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.

Università di Catania

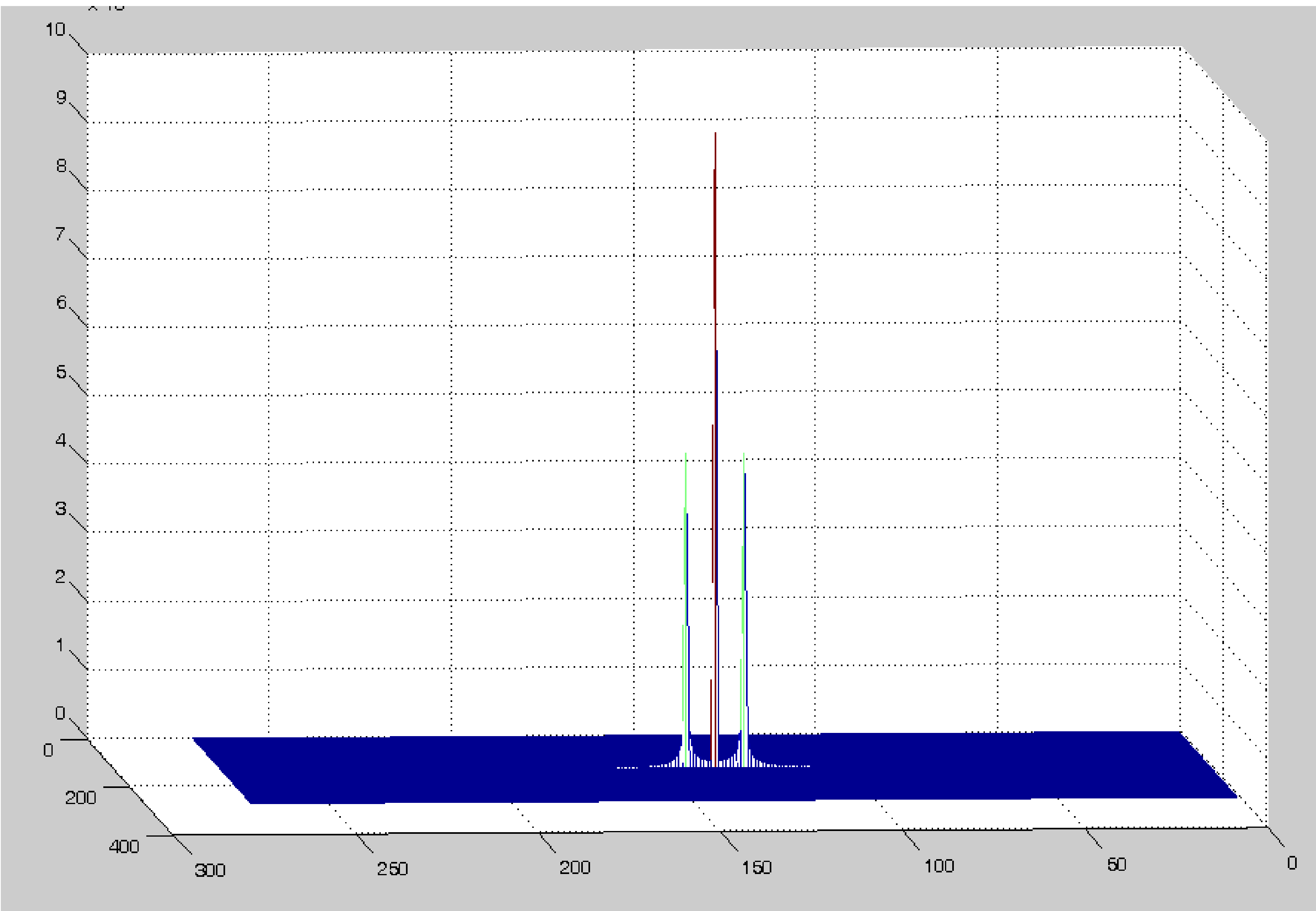
3D module



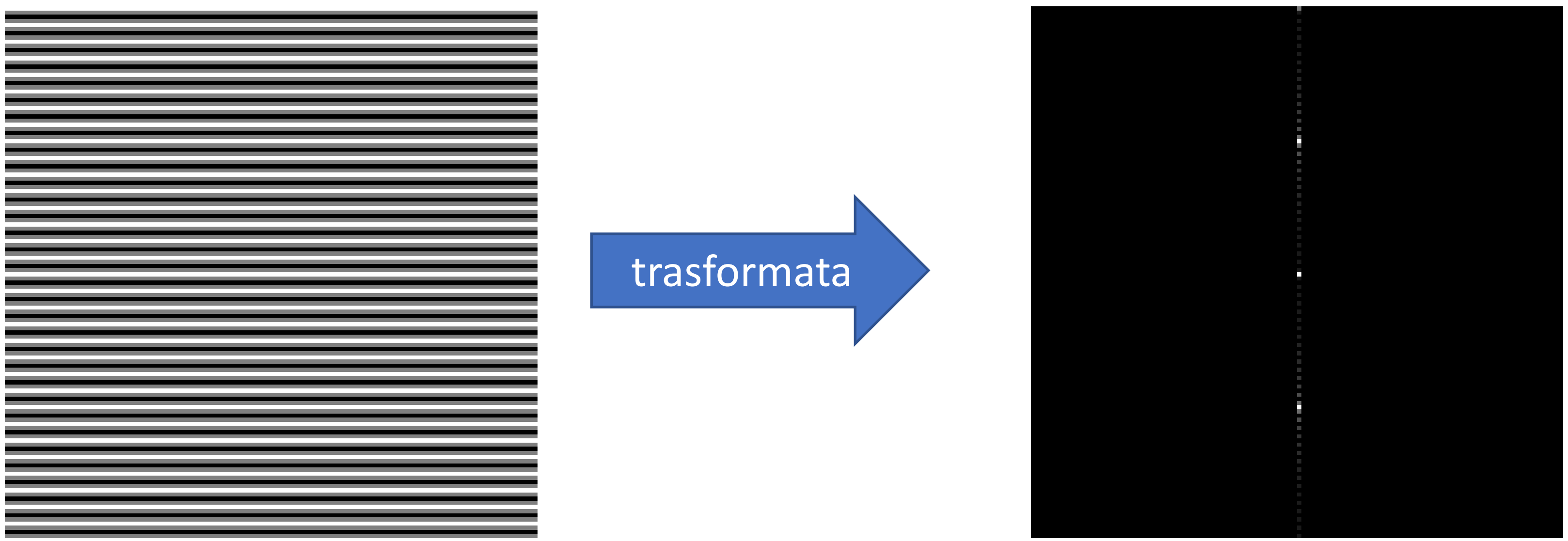
A few examples



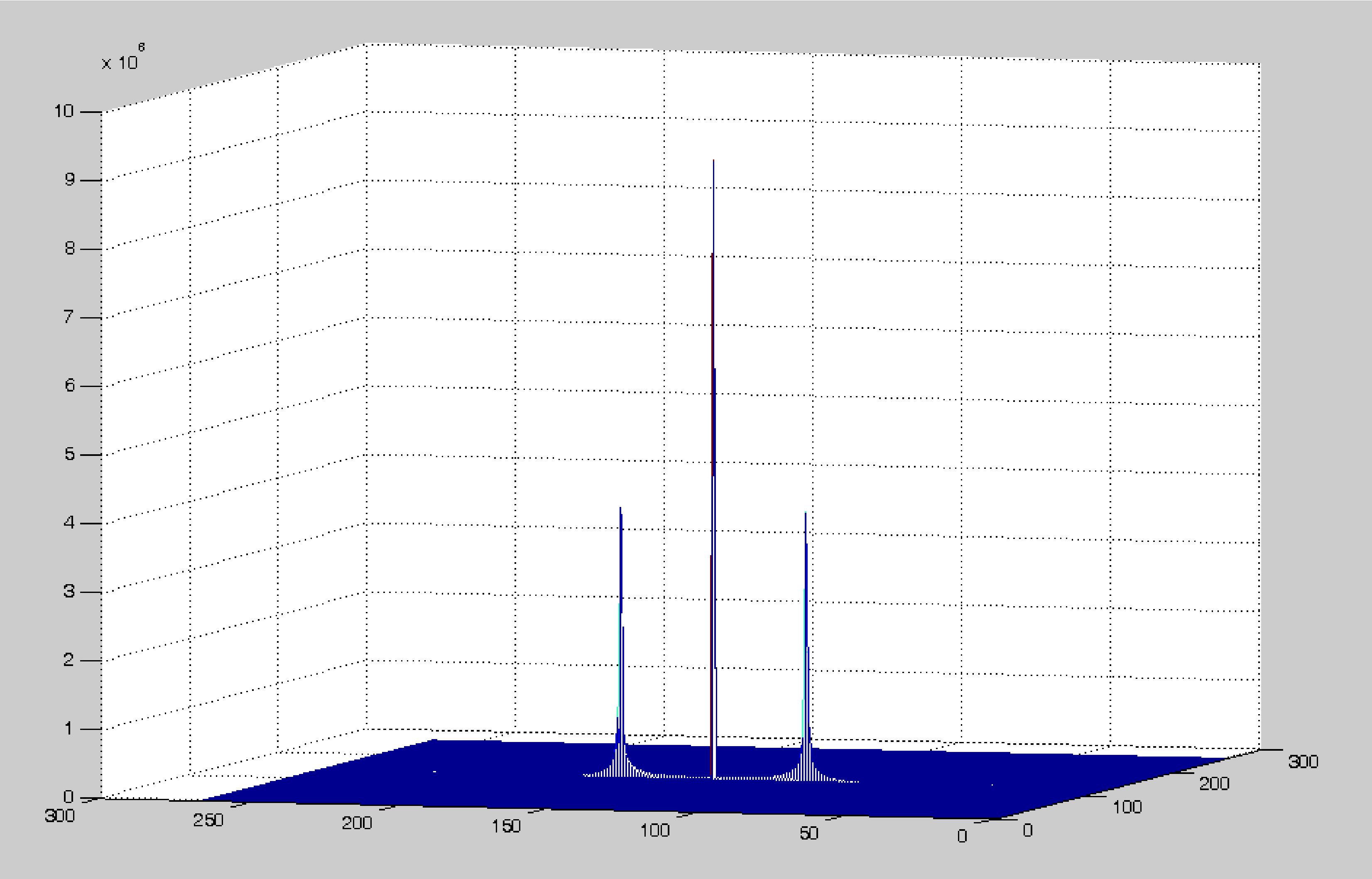
The spectrum in 3D



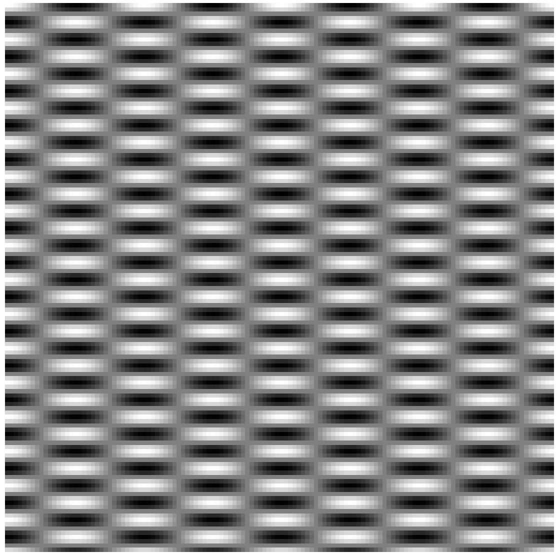
Other example



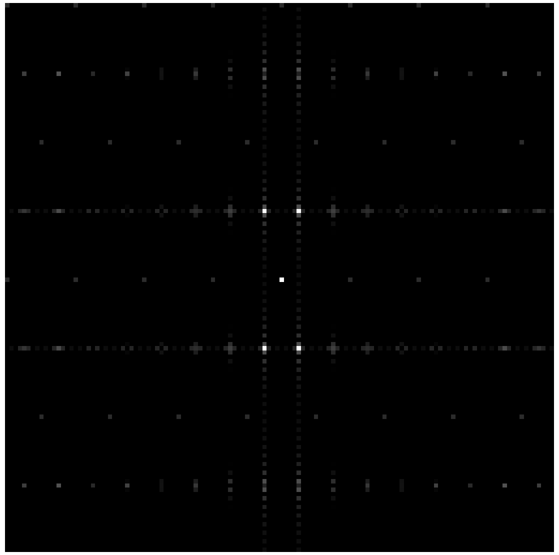
The spectrum in 3D



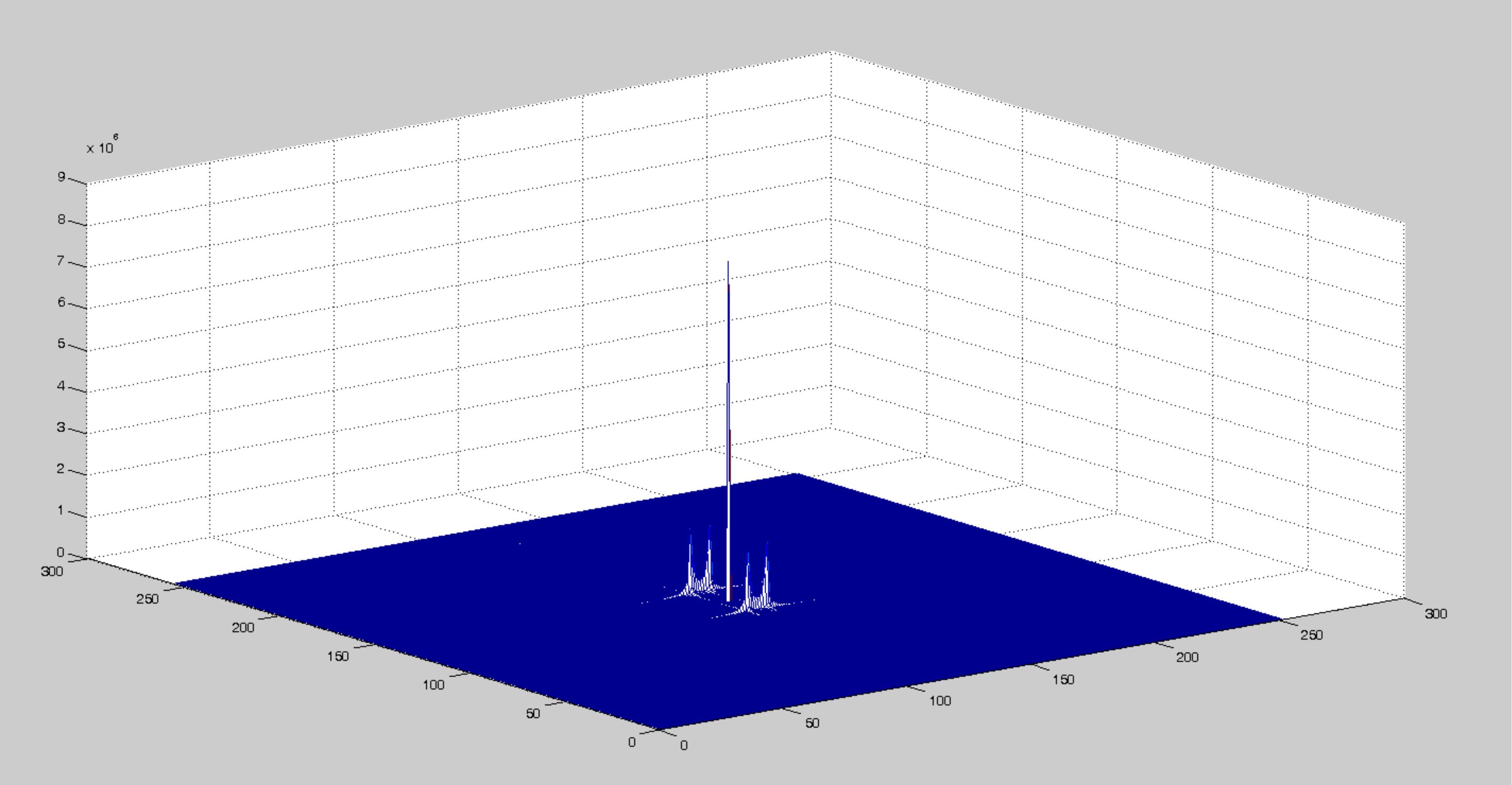
Example



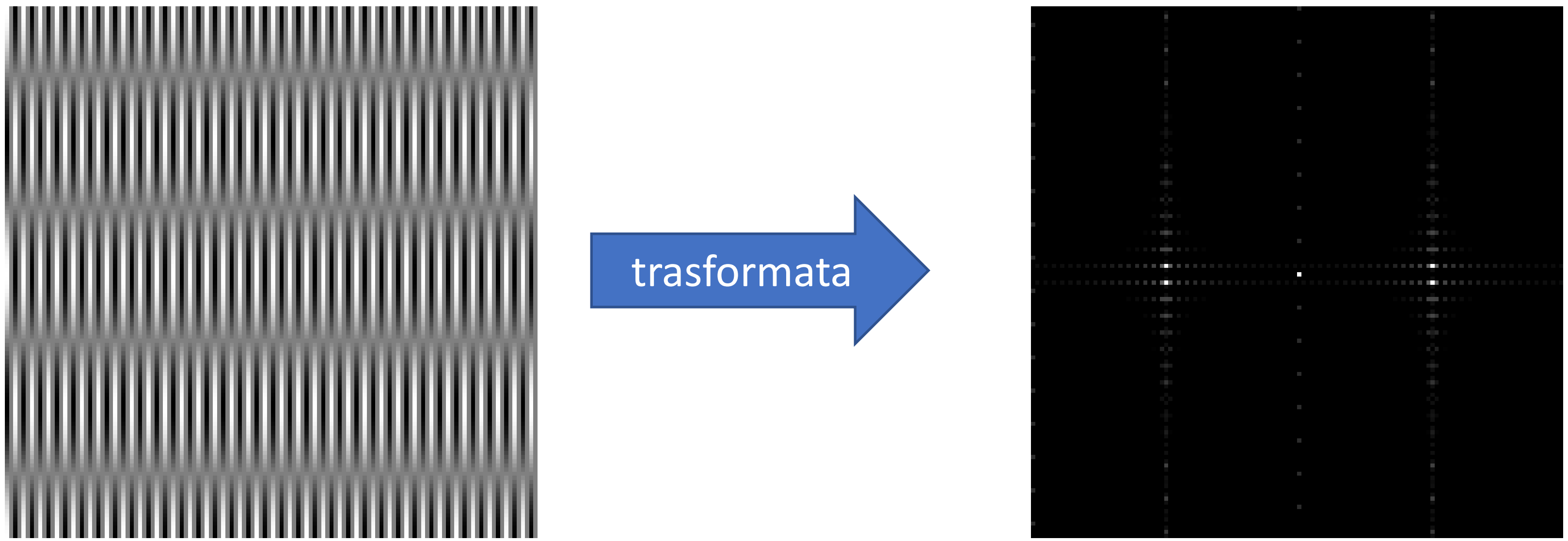
trasformata



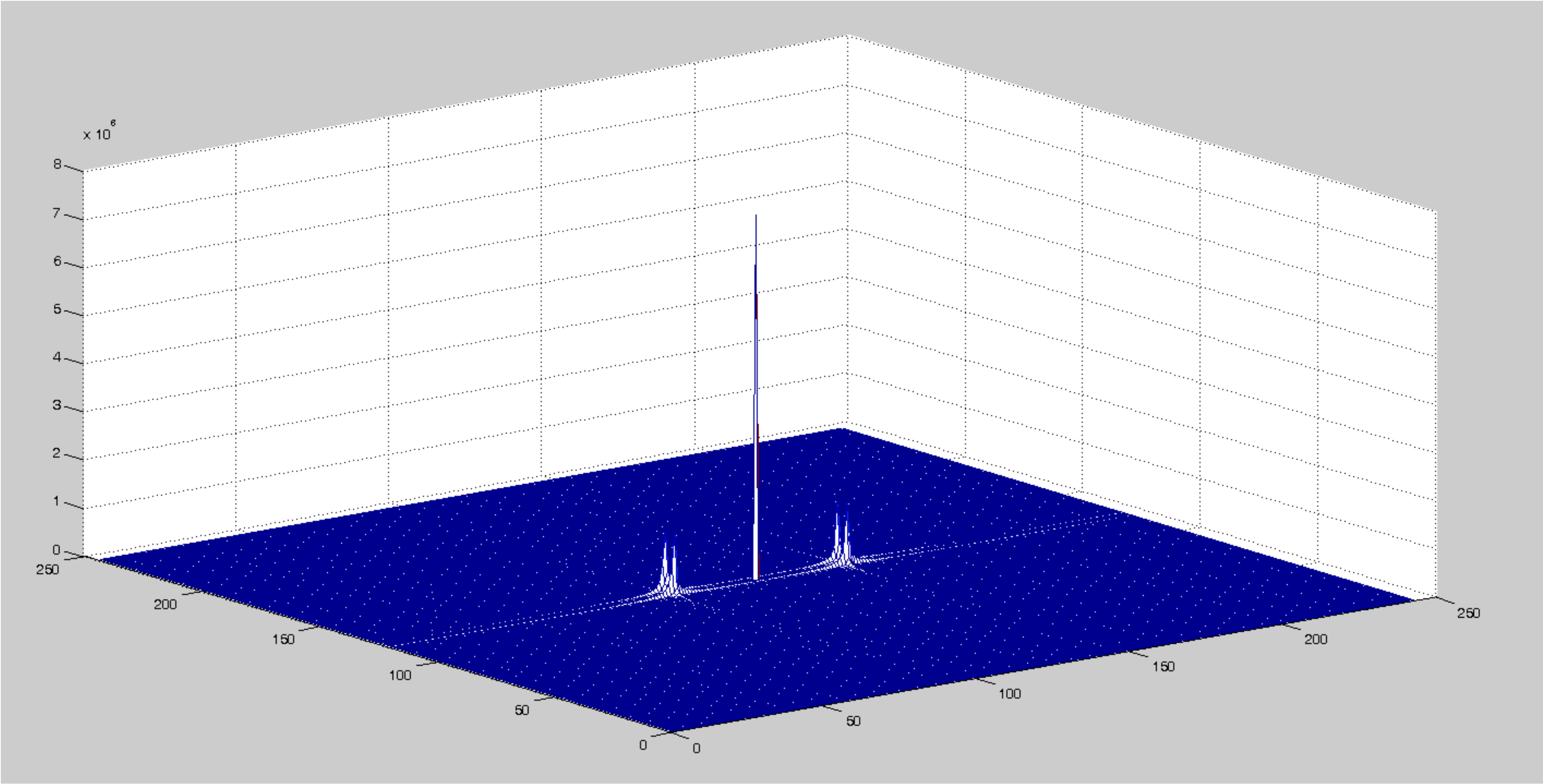
The spectrum in 3D



Example



The spectrum in 3D



Fourier transforms: advantages

What advantage can be obtained from the Fourier transform?

- In the space of frequencies it is possible to:
  - remove unwanted frequencies
  - reduce the space used by data while limiting signal degeneration (JPEG, MPEG, DivX, MP3)
  - regenerate degraded signals

Other Transforms

In addition to the Fourier transform, several transforms used in image processing, with extensive use in restoration and, especially, compression, belong to the class of unitary transforms.

These include:

- The discrete Walsh transform (DWT)
- The discrete Hadamard transform (DHT)
- The discrete Cosine transform (DCT)
- The discrete Karhunen Loeve transform (KLT)

Some properties of 2-D DFT

Image Processing Laboratory

Università di Catania

Some properties of 2-D DFT

- Separability
- Translation
- Mean Value

Separability

The discrete Fourier transform can be expressed in separable form. In particular, the following expression applies:

$$F(u,v) = \frac{1}{M}\sum_{x=0}^{M-1} g(x,v) e^{\frac{-i\pi ux}{M}}$$

where:

$$g(x,v) = \left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y) e^{\frac{-i2\pi vy}{N}} \right]$$

The main advantage of the separability properties is that F(u,v) can be obtained by applying the 1-D transform in two successive steps.

Translation

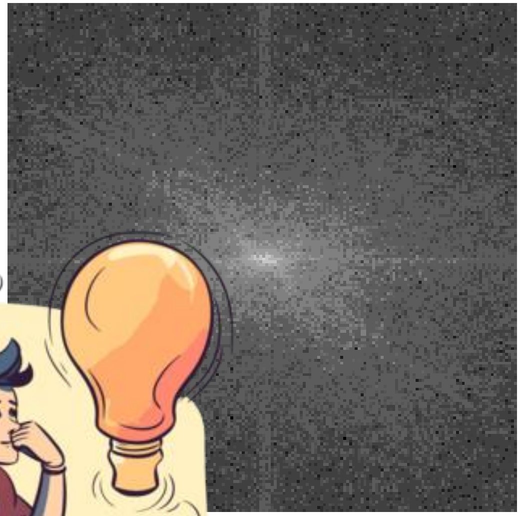
In the two-dimensional case it is useful before operating on the transform to apply a shift (translation) of the origin at the point (M/2, N/2) that is, the center of the coefficient matrix of frequencies.

This shifts the data in such a way that F(0,0) results in the center of the rectangle of frequencies defined between [0,M-1] and [0,N-1].

It is also shown that a shift in f(x,y) does not change the magnitude of the transform.

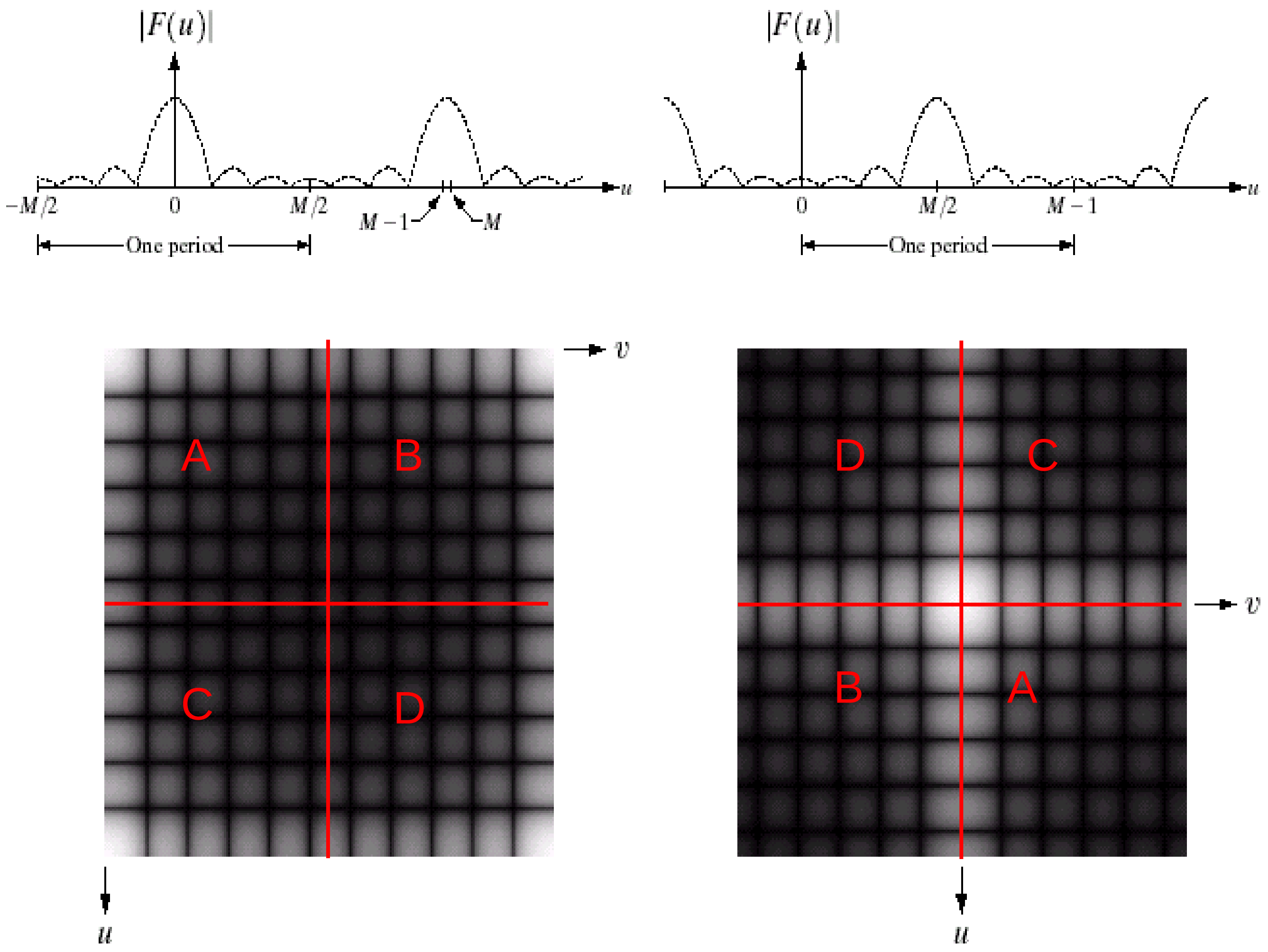
These properties are used for better visualization of the spectrum.

$f(x,y)$



$|F(u,v)|$

Translation



Mean Value

The value of the transform at the origin, that is, at the point (u,v)=(0,0) is given by:

$$F(0,0) = \frac{1}{NxN} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \qquad \bar{f}(x,y) = \frac{1}{NxN} F(0,0)$$

As can be seen, it is nothing but the mean of f(x,y). The value of the Fourier transform of an image f(x) in the origin is equal to the average of the gray values contained in the image.

F(0,0) also takes the name of continuous component or DC component.

Fast Fourier Transform

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-i2\pi ux/N]$$

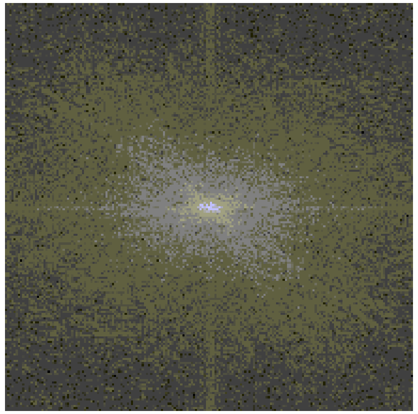
- In its classical form implementing the Fourier transform would require a number of operations proportional to N^2 (N complex multiplications and N-1 additions for each of the N values of u).

- Using appropriate decomposition techniques, it is possible to lower the complexity to $N\log_2 N$ by implementing the so-called Fast Fourier Transform (FFT).

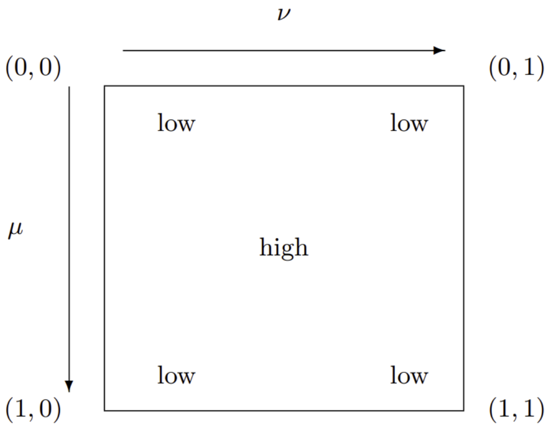
Frequencies: Low and High

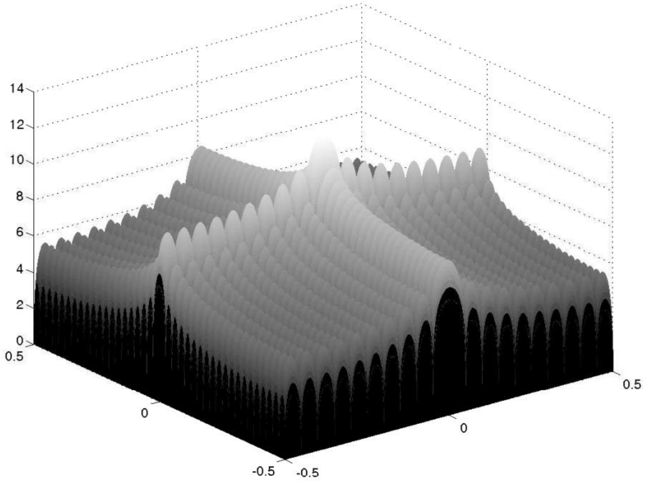
- Normally **it is impossible to make direct associations between specific parts of the image and its transform** (loss of spatial localization).

- Remembering that frequency is related to rate of change, however, it is possible to associate **low frequencies with uniform areas** of the image, **high frequencies with more or less abrupt changes and thus with edges or noise**.



$D(u,v)$



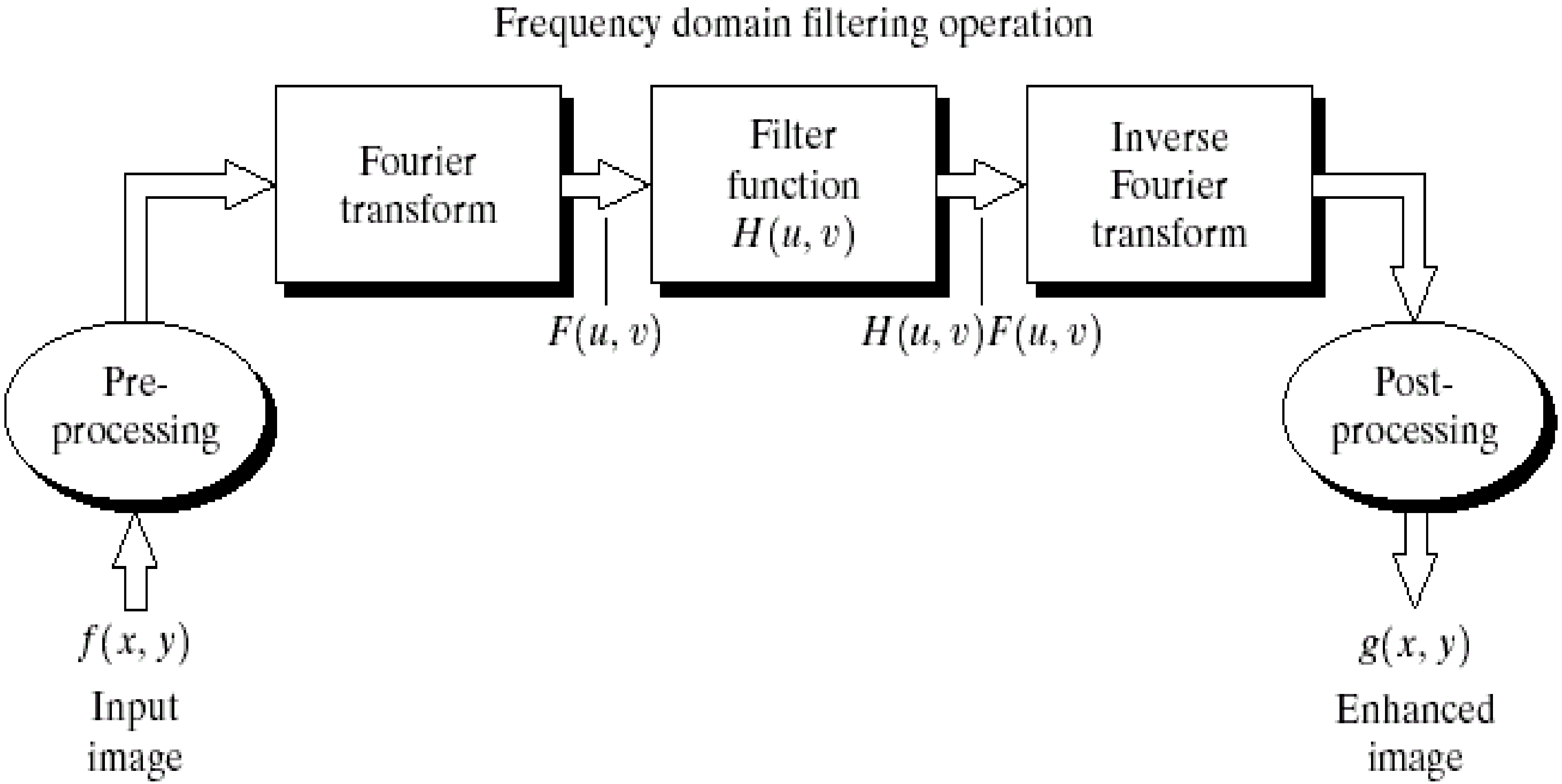


Filtering in the Frequency Domain

Luca Guarnera

An introduction on Deepfakes Creation and Detection Approaches

Filtering in the Frequency Domain

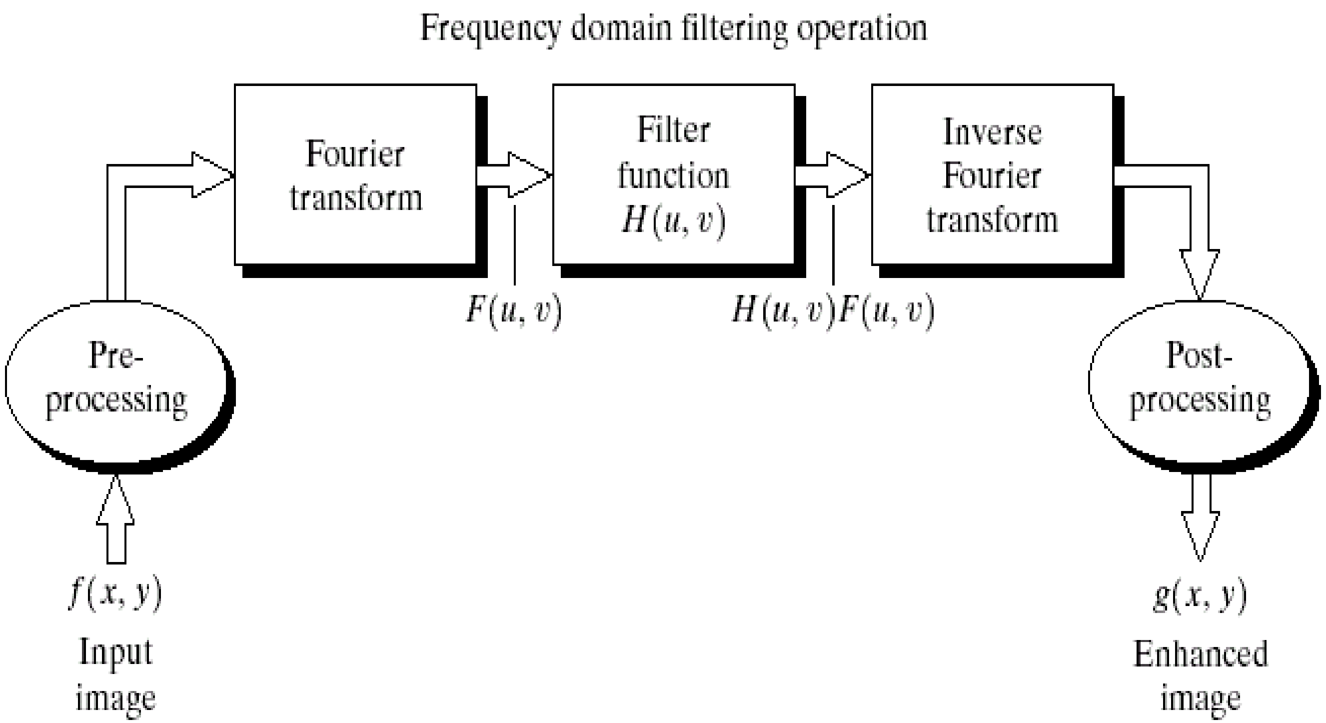


Filtering in the Frequency Domain

- The function **H(u,v)** is called a **filter** because it **acts on some frequencies of the transform while leaving the others unchanged**.

- **Very often the H function is a real function and each of its components multiplies both the corresponding real and imaginary components of the F.**

- This type of filter is called ***zerophaseshift*** because it does not introduce phase shifting.



Convolution Theorem

Why is the frequency domain and not the spatial domain used to use **global operators**?

Because the following theorem applies:

The convolution of two signals in the spatial domain is equivalent to the antitransform of the product of frequencies.

Convolution Theorem

- The theoretical foundation of processing techniques in the frequency domain, based on DFT manipulation of the image, is represented by the **convolution theorem**, which makes the operation thus defined correspond to the spatial domain:

$$g(x,y) = f(x,y) * h(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m, y-n)$$

- the operation, in the frequency domain:

$$G(u,v) = F(u,v)H(u,v)$$

Convolution Theorem

- So if the convolution operation in the spatial domain is defined as follows:

$$g(x,y) = f(x,y) * h(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$$

- The same operation in the frequency domain becomes:

$$g(x,y)=F^{-1}\{F(u,v)H(u,v)\}$$

Convolution Theorem

- Complexity for a 1D signal:
  - In the frequency domain O(n logn)
  - In the spatial domain O(n2)

- It is actually worthwhile to move to the frequency domain!

Filtering in the Frequency Domain

- If the filter is comparable in size to the image, it is more computationally efficient to perform filtering in the frequency domain.

- With smaller masks it becomes more computationally efficient to compute in the spatial domain.

- Defining a filter in the frequency domain is more intuitive.

How to get a filter from a spatial mask

1. The filter H has the same size as the image I;

2. H must have in the upper left corner the values of the spatial mask, in the rest always the value 0;

3. A shift of H is made.

4. One calculates from H the fourier transform.

Low pass filters in the frequency domain

Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Low pass ideal

$$H(u,v) = \begin{cases} 1 & \text{se } D(u,v) \le D_0 \\ 0 & \text{se } D(u,v) > D_0 \end{cases}$$

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

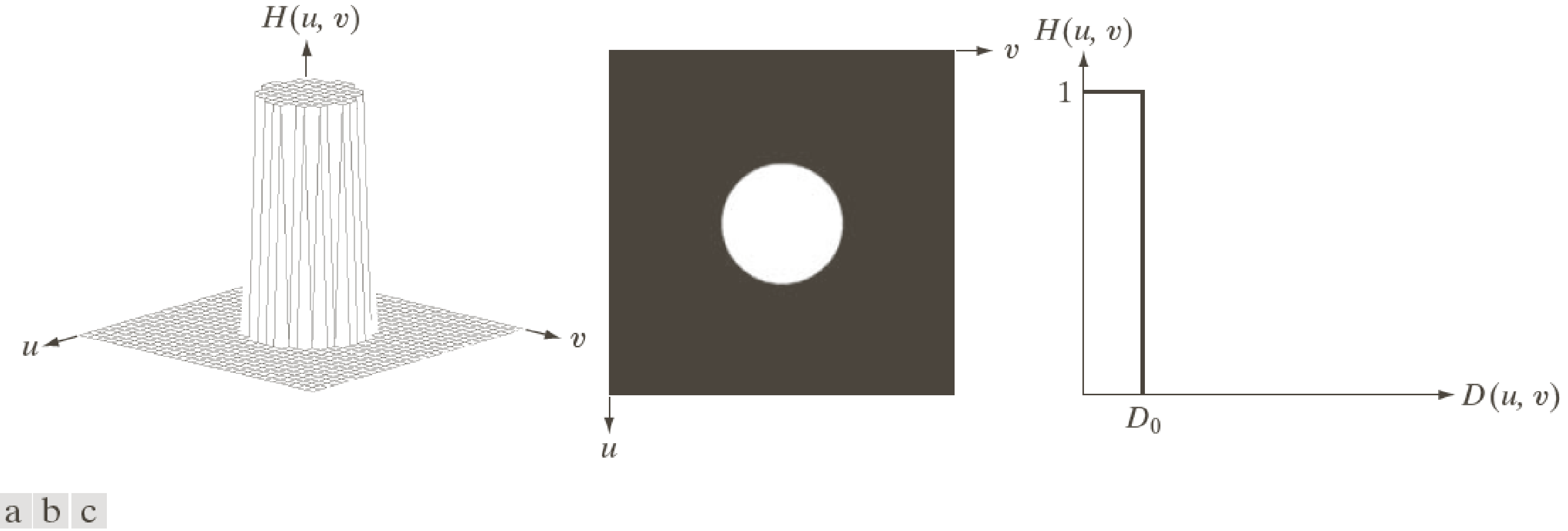
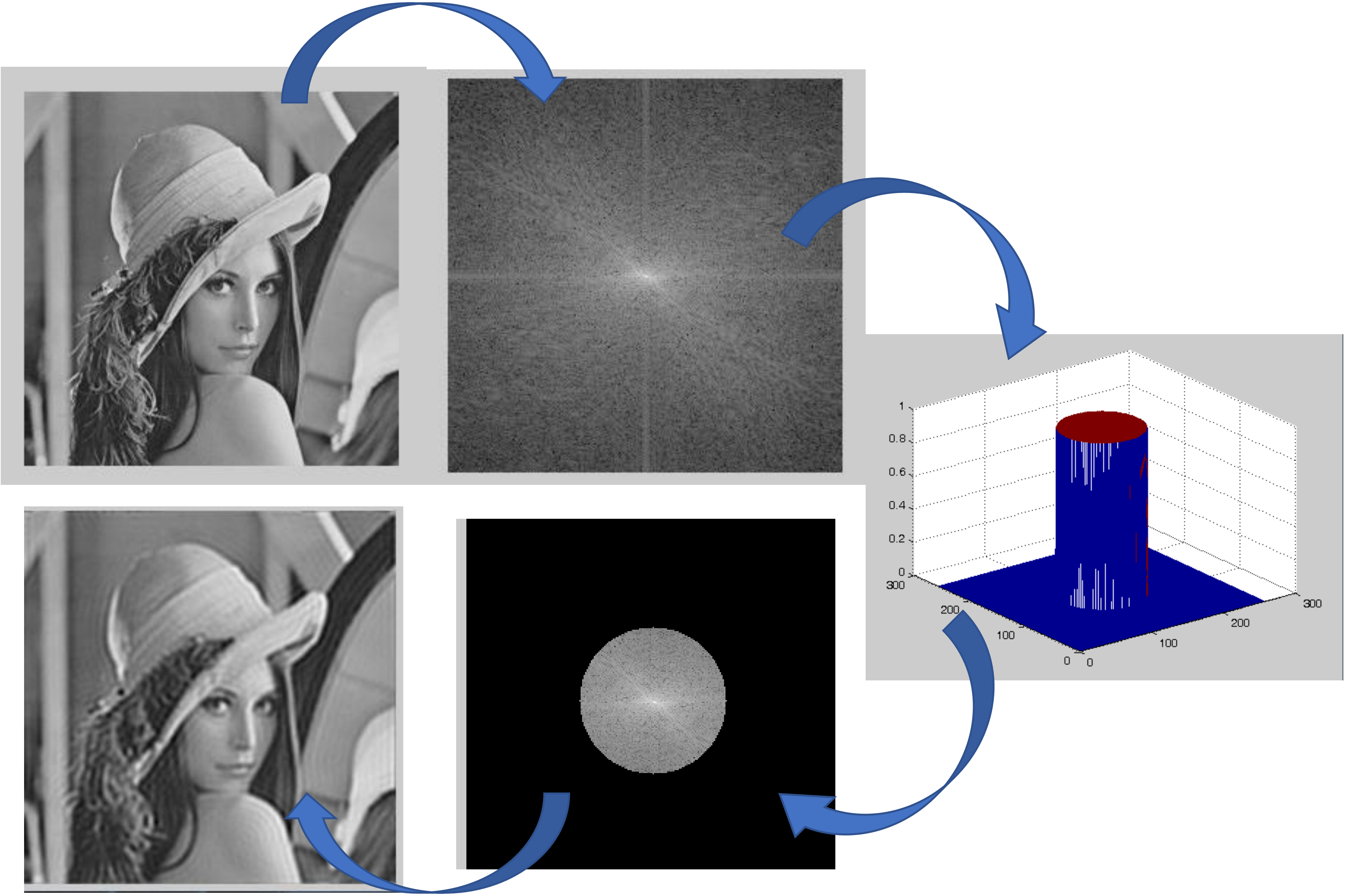


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

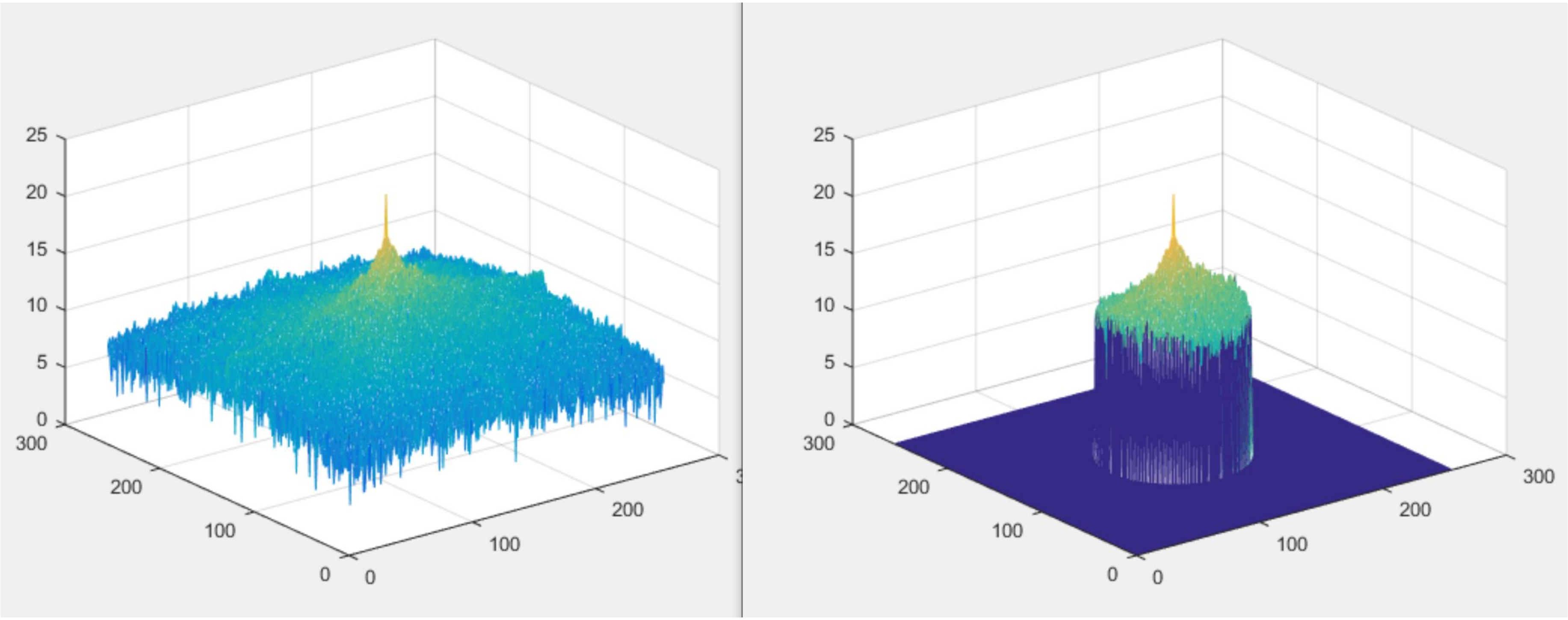
Ideal low-pass filter



Ideal low-pass filter with D_0=50

$$H(u,v) = \begin{cases} 1 & \text{se } D(u,v) \le D_0 \\ 0 & \text{se } D(u,v) > D_0 \end{cases}$$

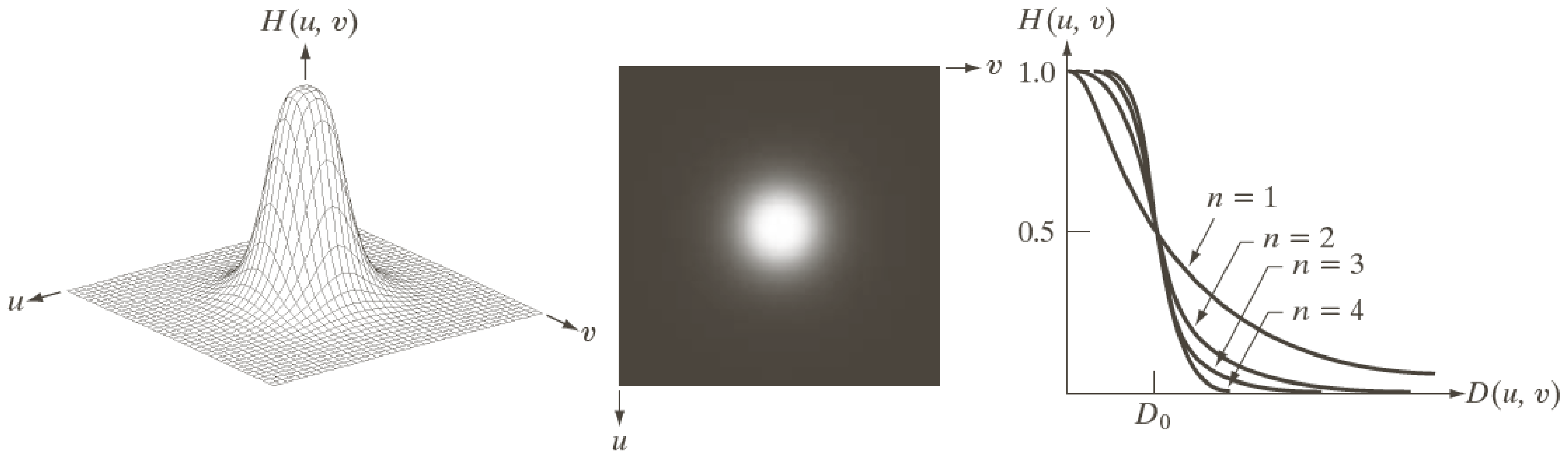
$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2\right]^{1/2}$$



Butterworth low pass filters

The transfer function of the **Butterworth low-pass filter** of order n and cutoff frequency D_0 è:

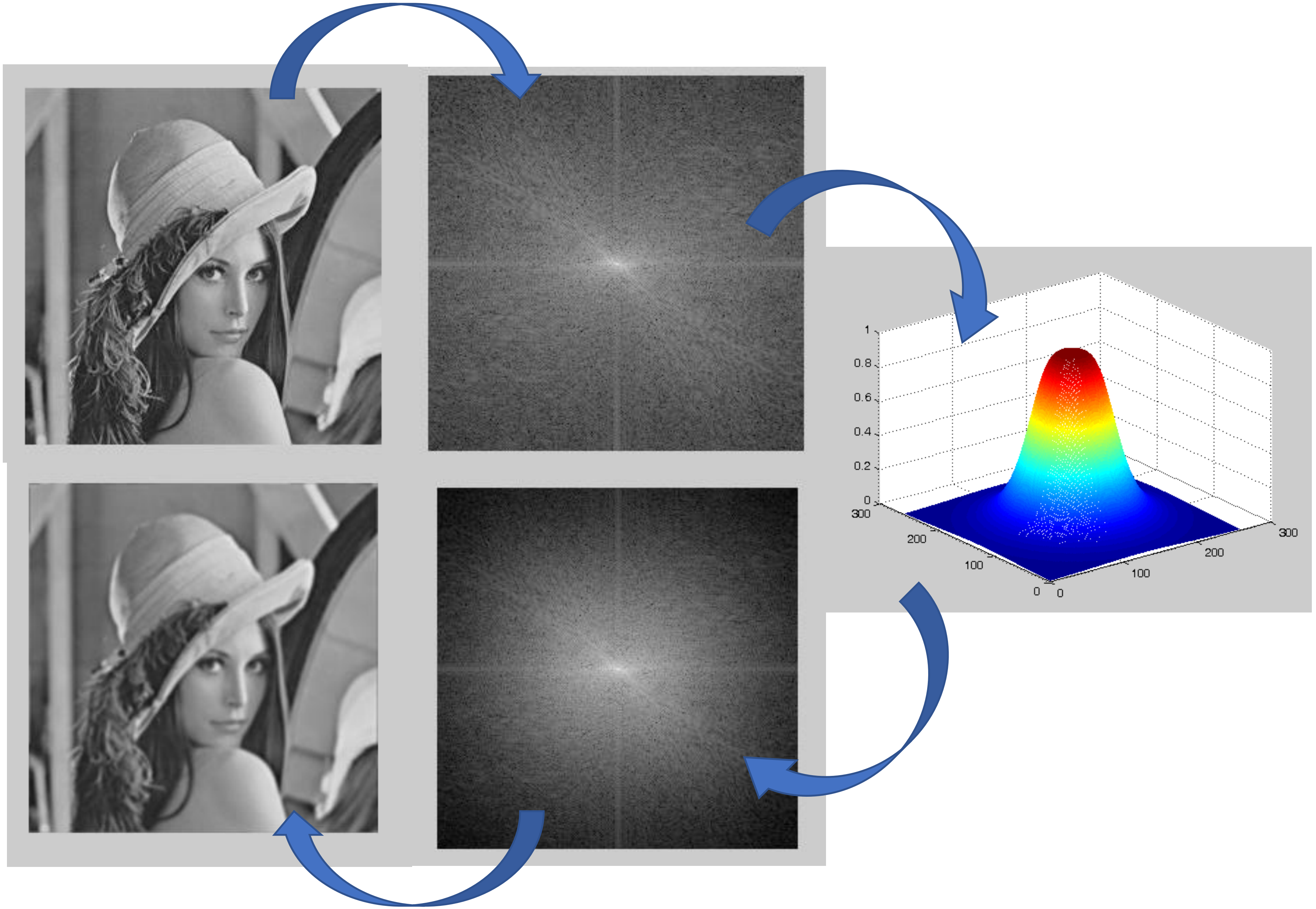
$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}}$$



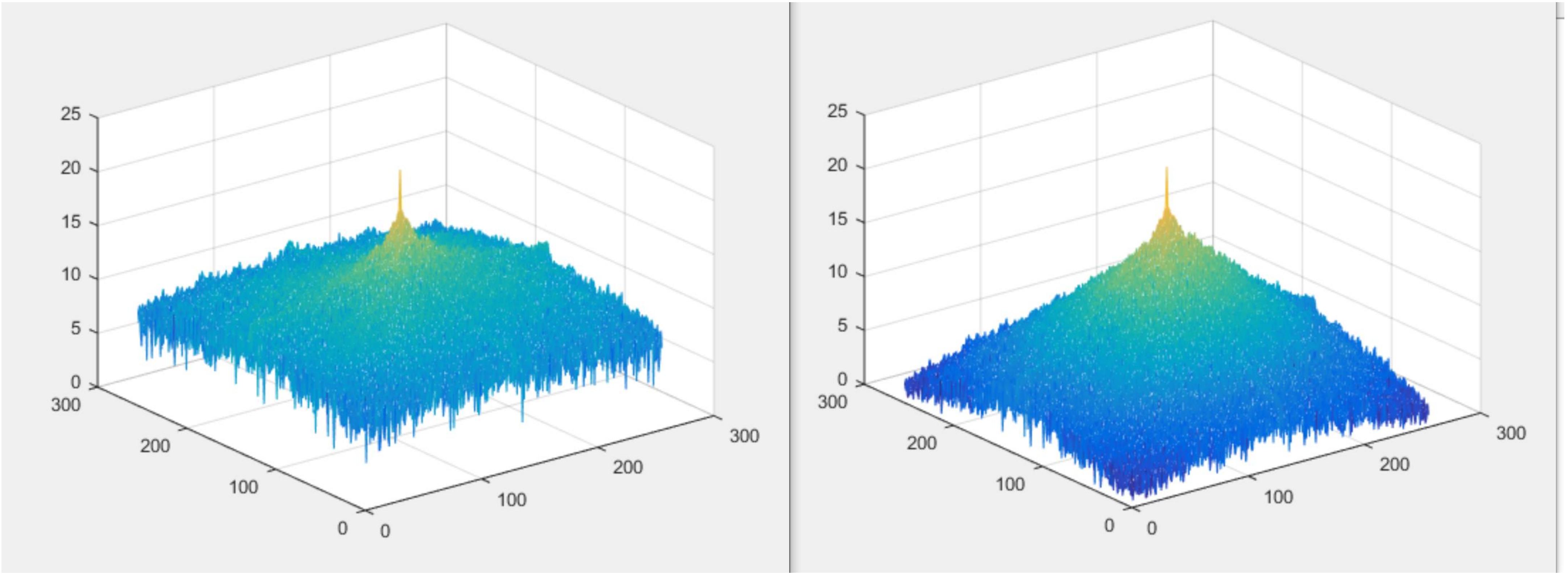
a b c

FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth low-pass filter

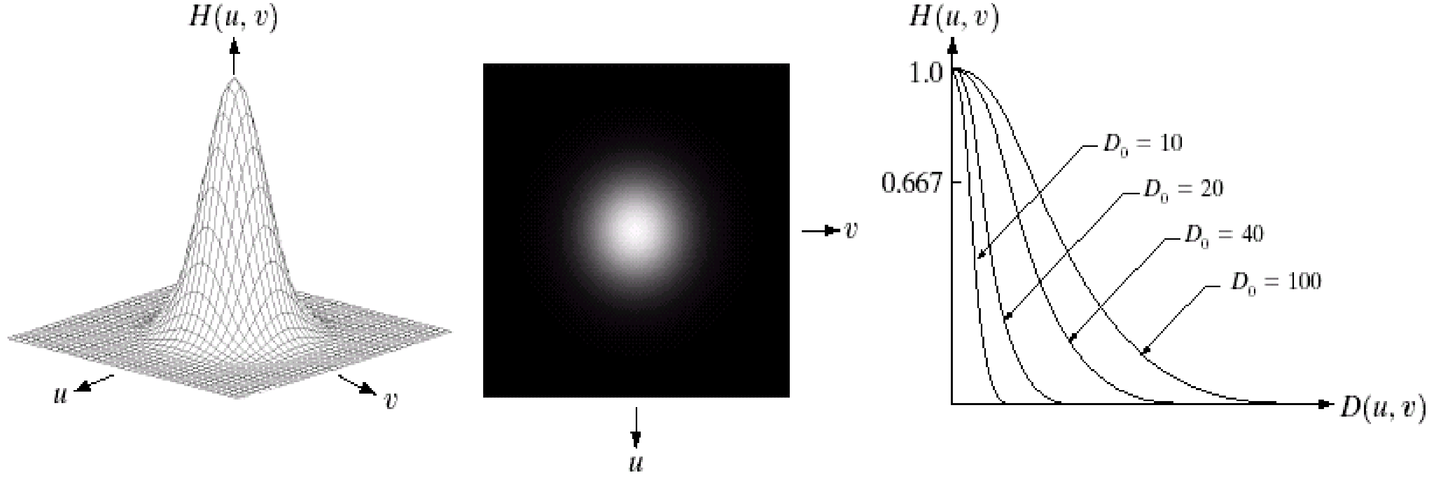


Butterworth low-pass filter with D_0=50

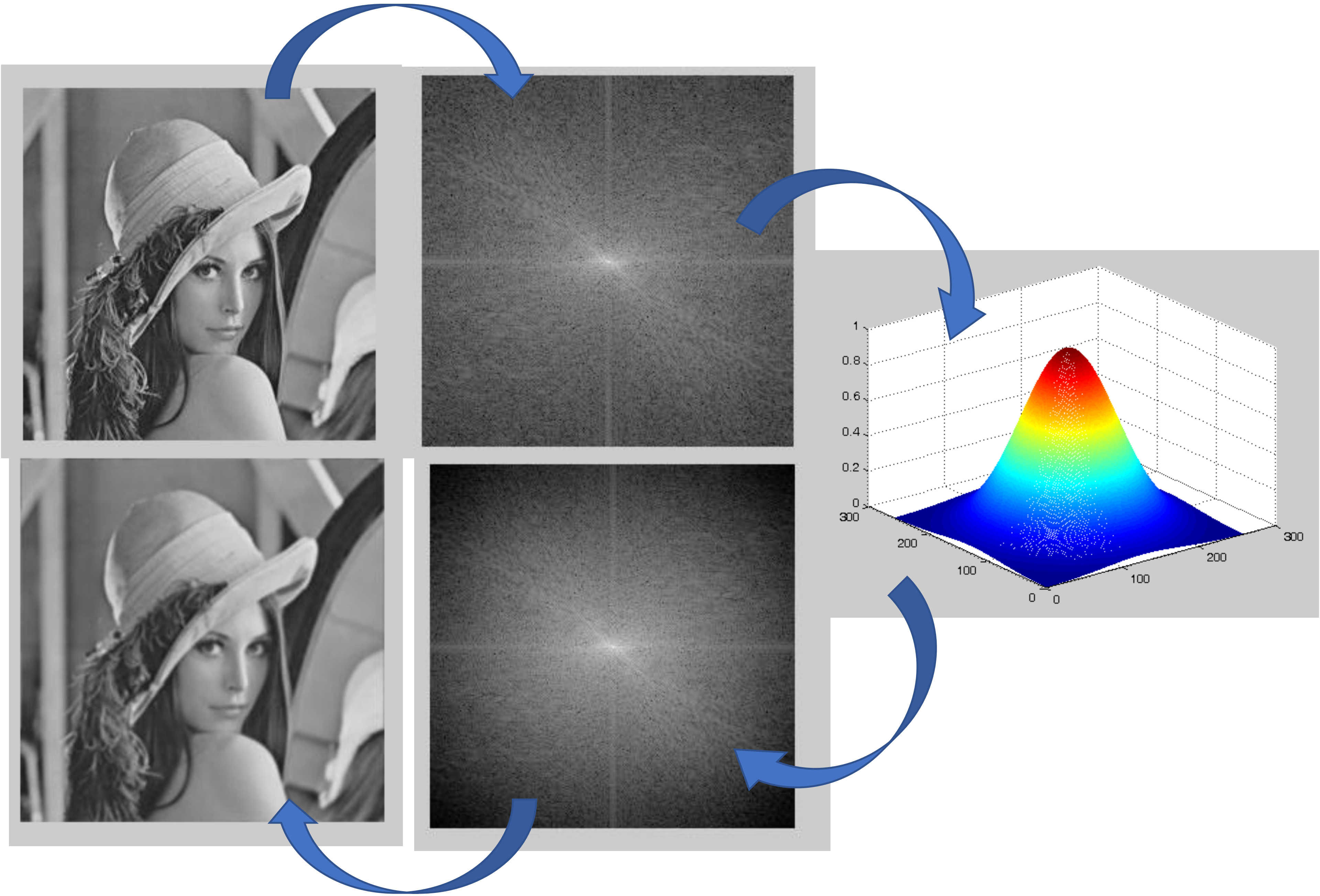


Gaussian filter

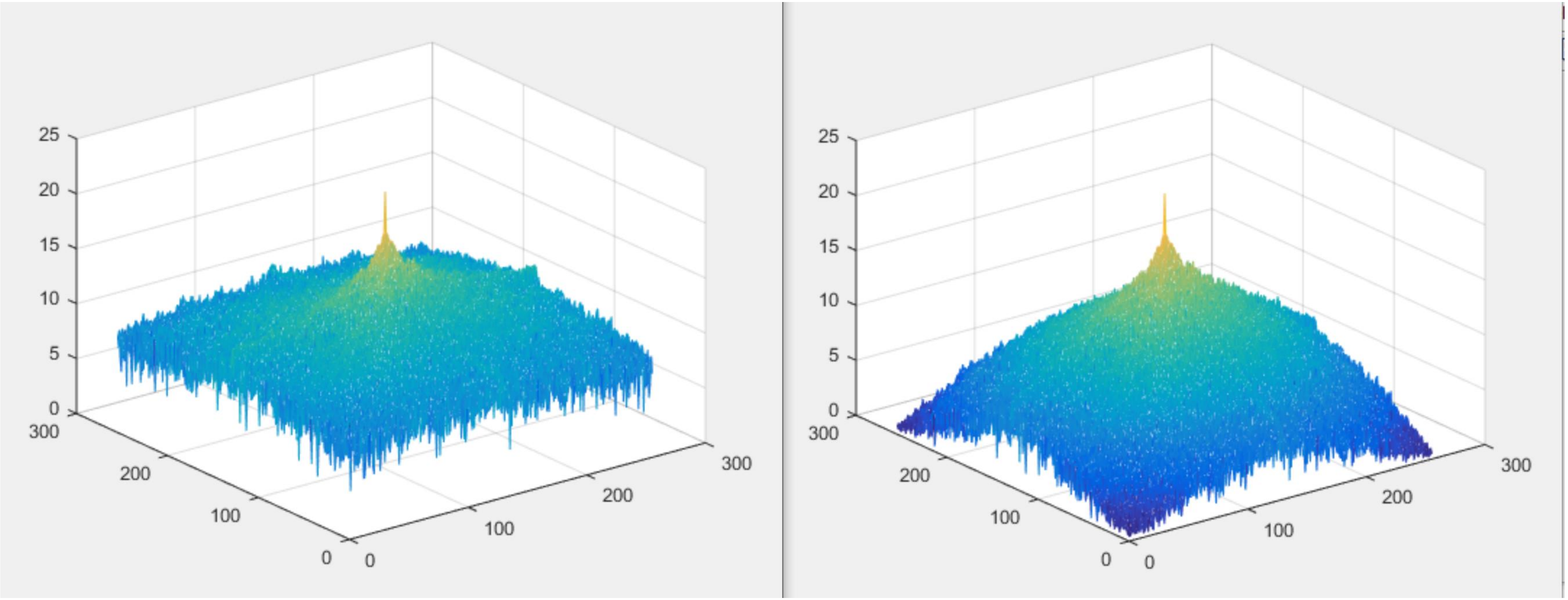
- Gaussian filters are defined by: $$H(u,v) = e^{\frac{-D^2(u,v)}{2D_0^2}}$$
- **Gaussian filters have the great advantage of still having a Gaussian as the Fourier transform.**



Gaussian low-pass filter



Gaussian low-pass filter with D_0=50

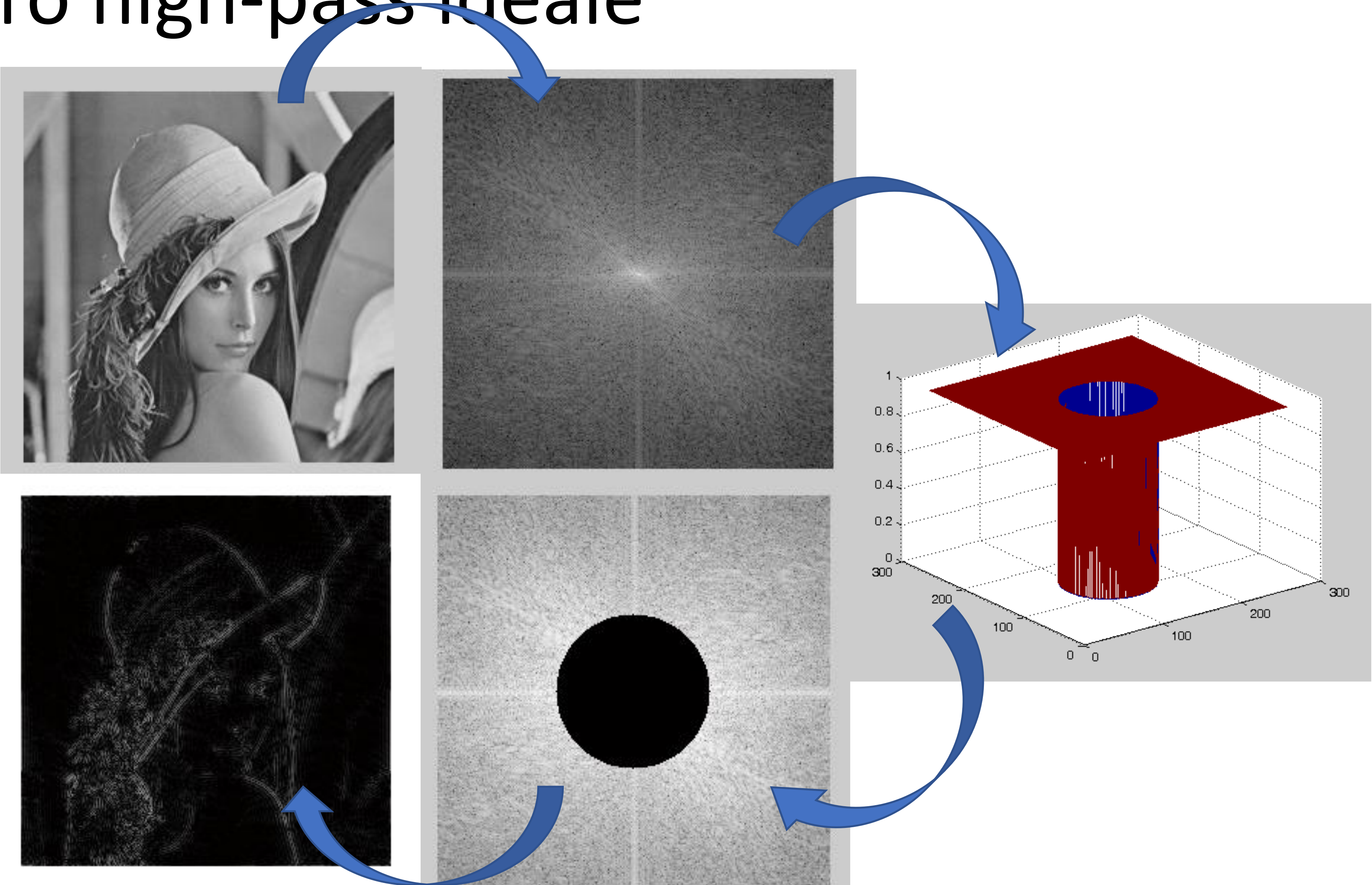


High pass filters in the frequency domain

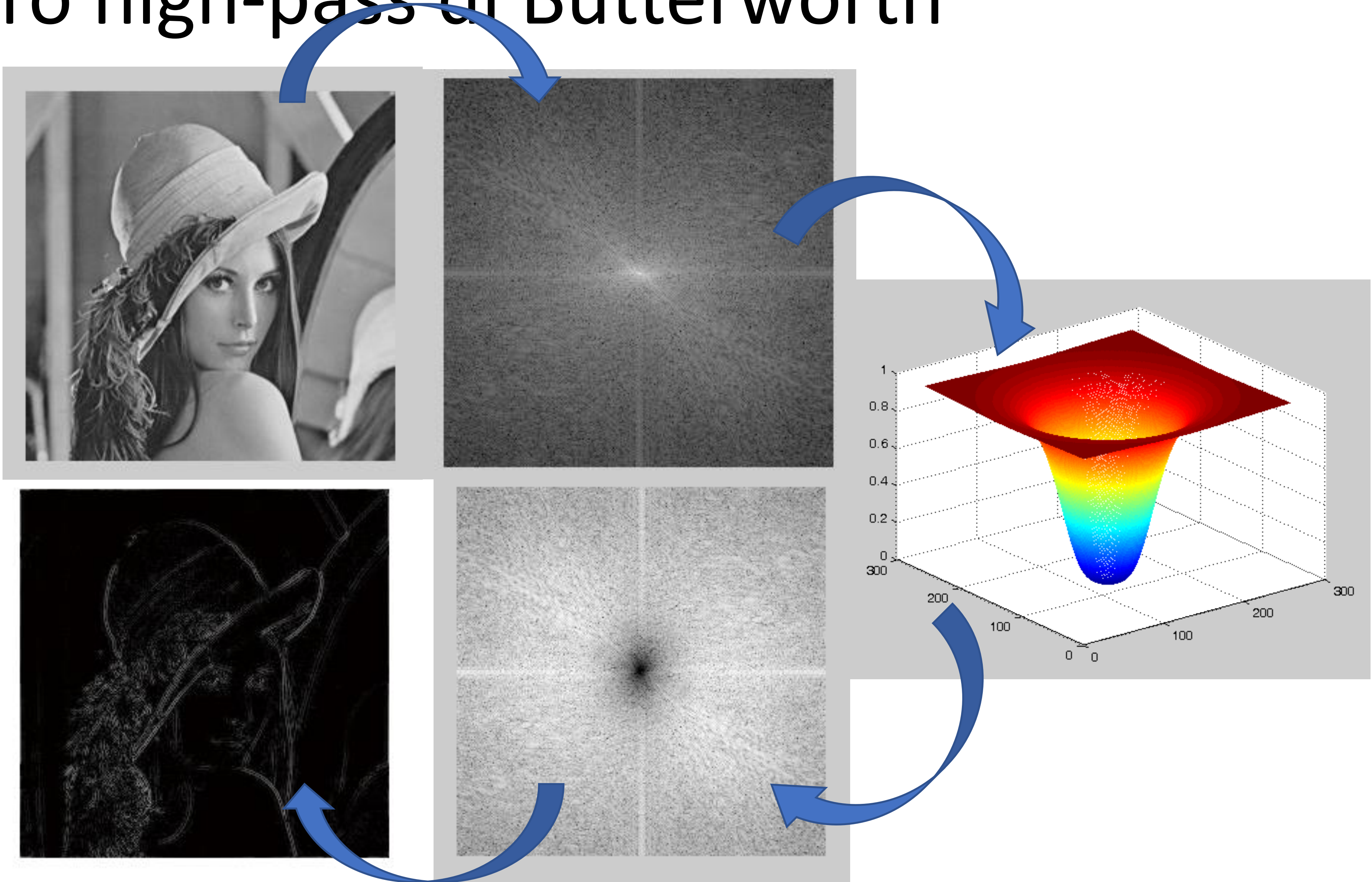
Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$

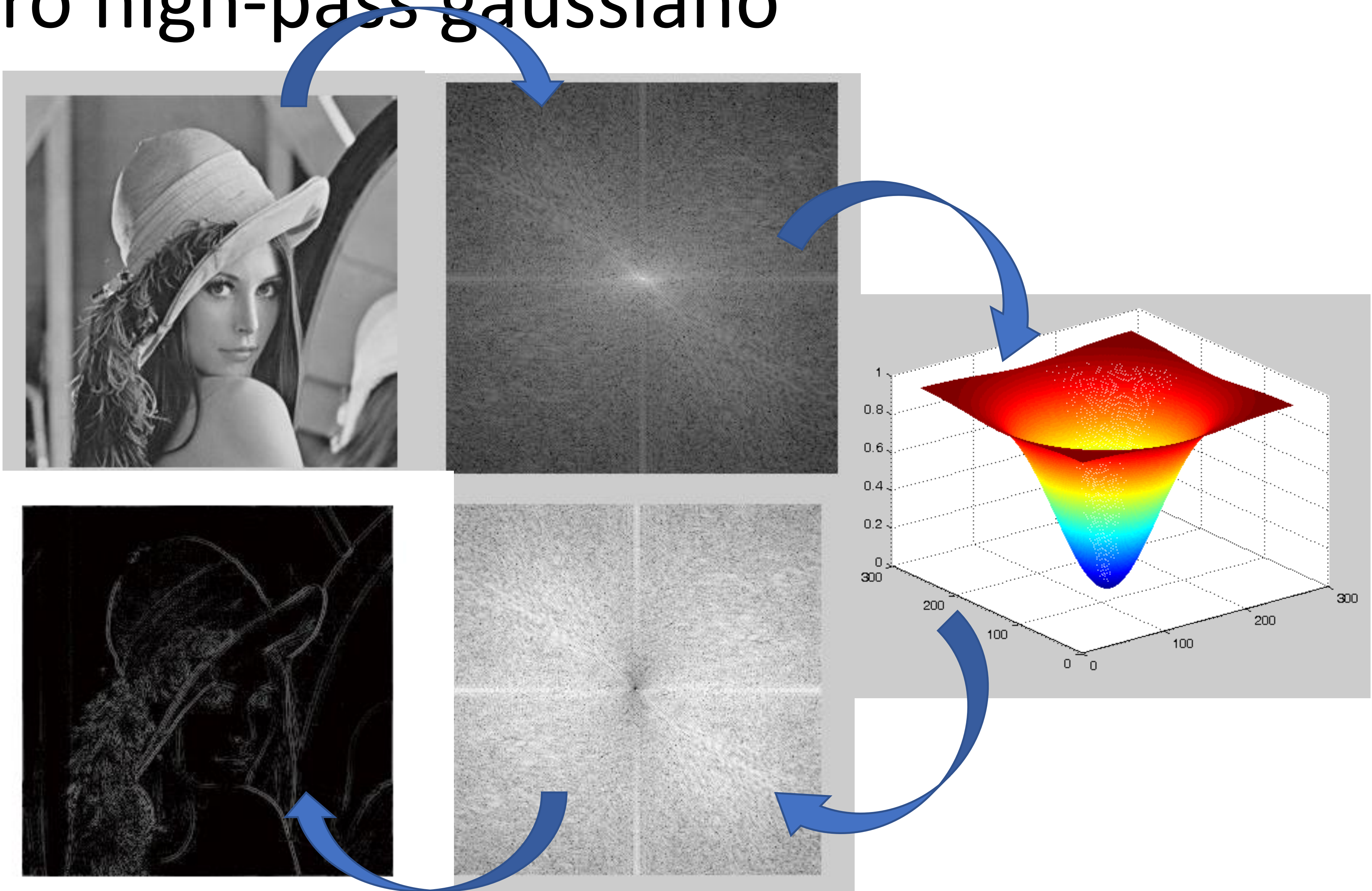
Filtro high-pass ideale



Filtro high-pass di Butterworth



Filtro high-pass gaussiano



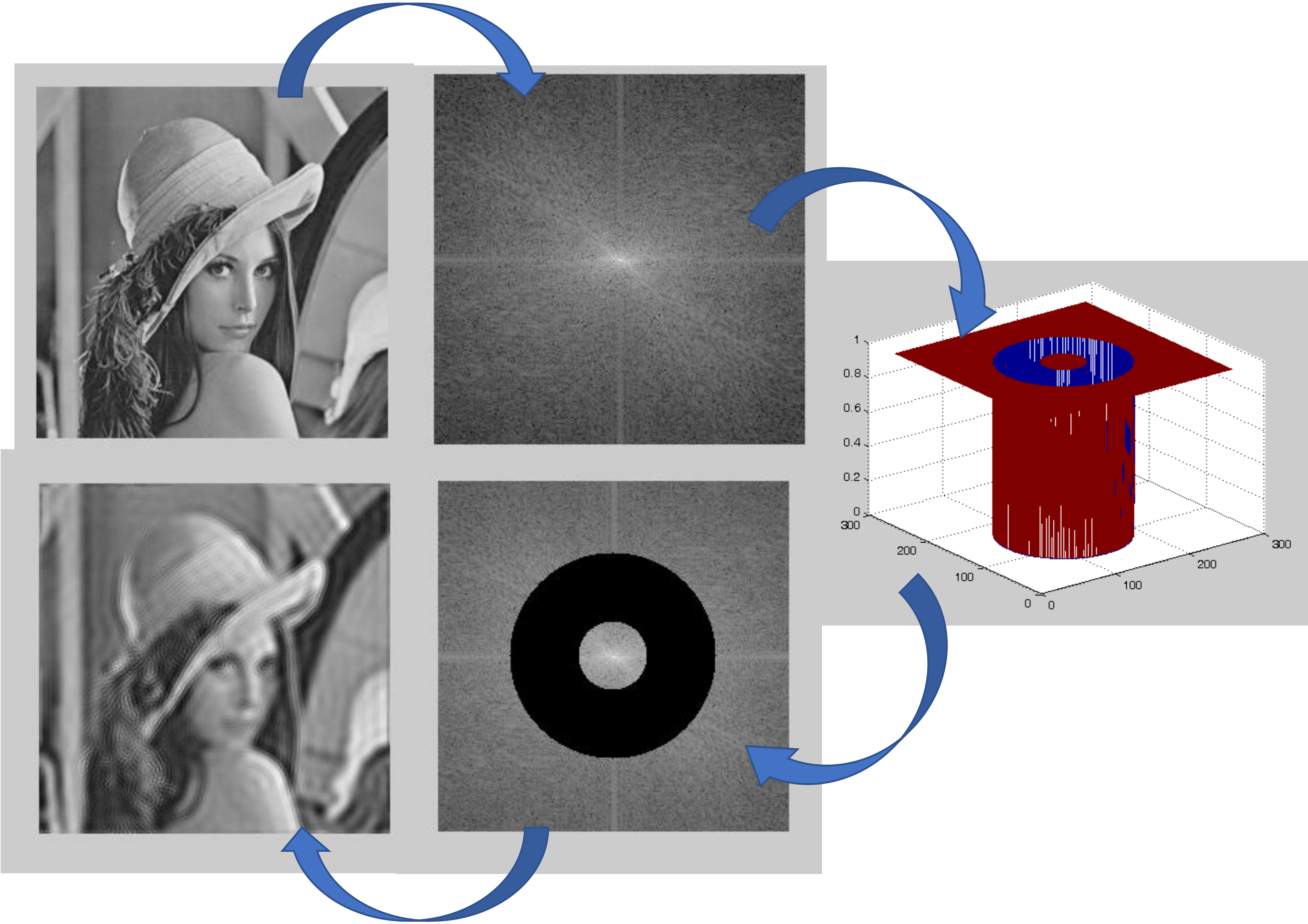
Band reject filters in the frequency domain

Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

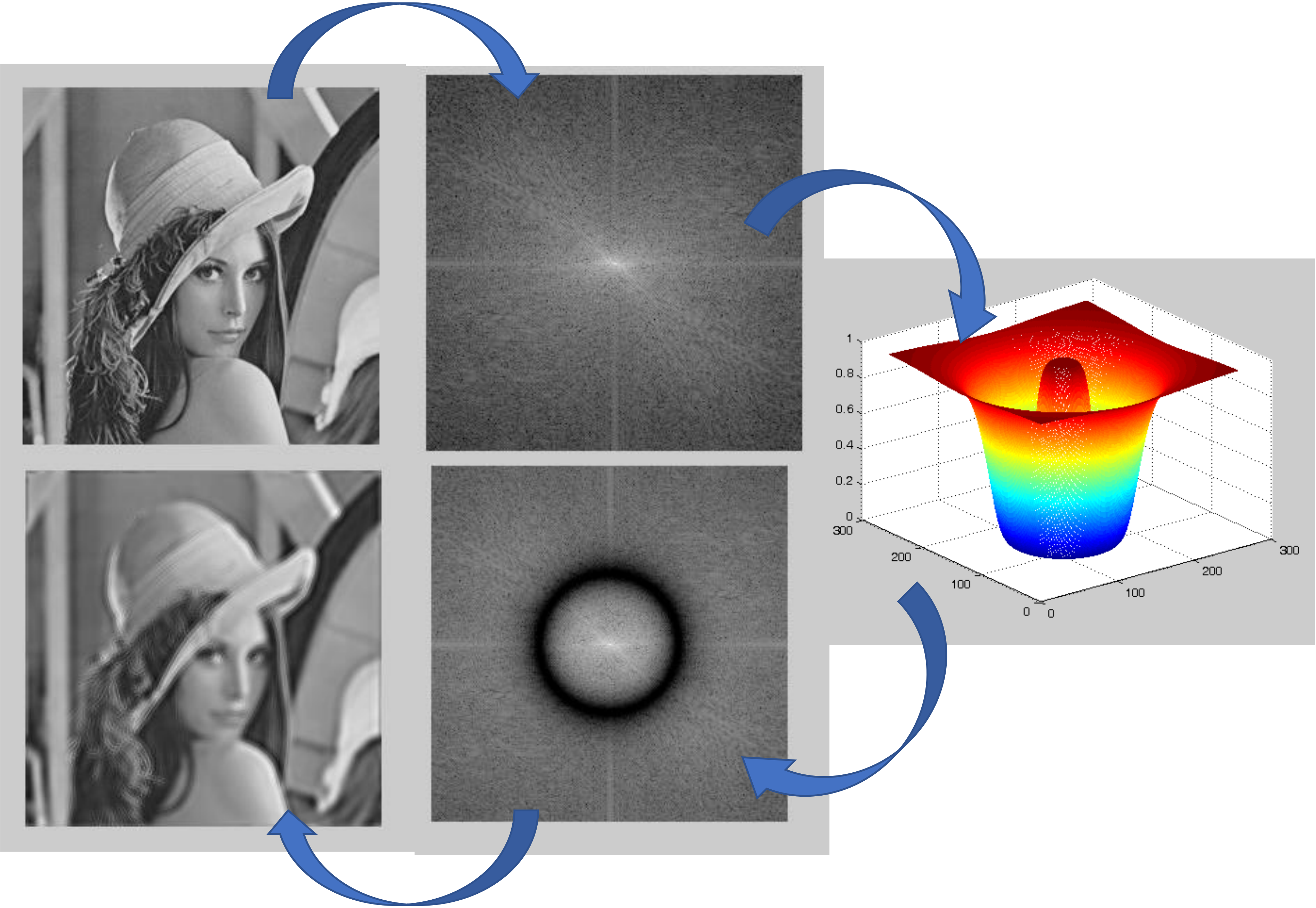
Ideal band-reject filter

$$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$$



Butterworth band-reject filter

$$H(u,v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2}\right]^{2n}}$$



Gaussian band-reject filter

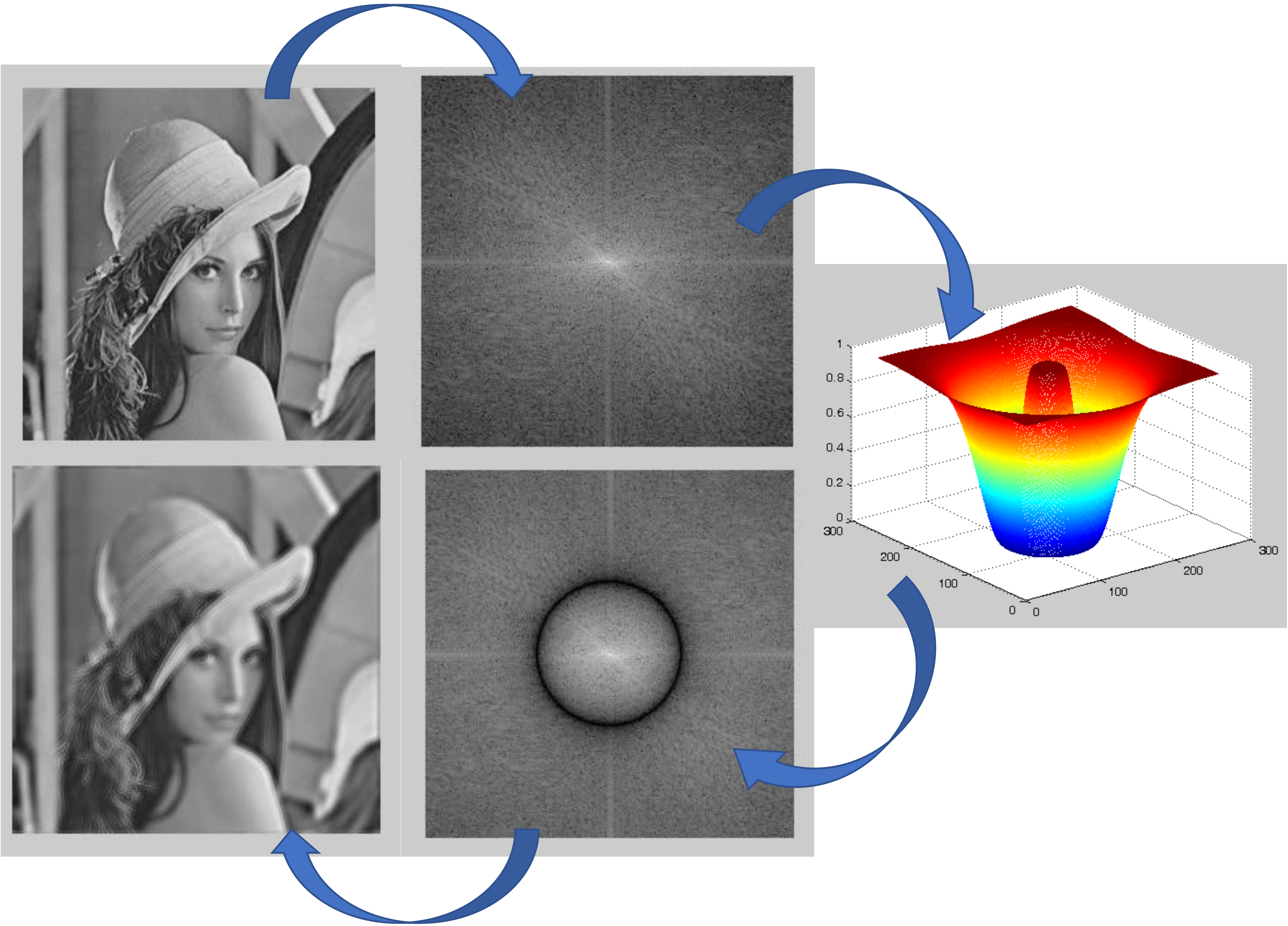


TABLE 4.4

Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

TABLE 4.5

Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

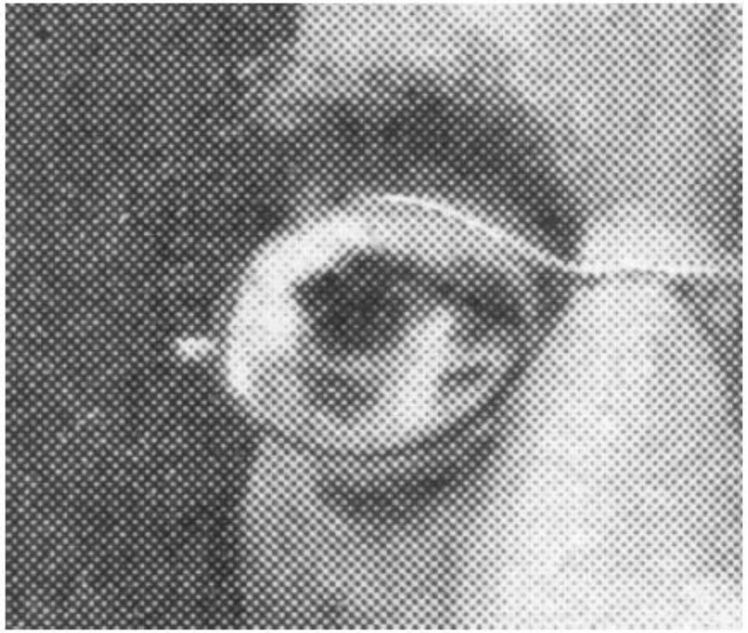
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$

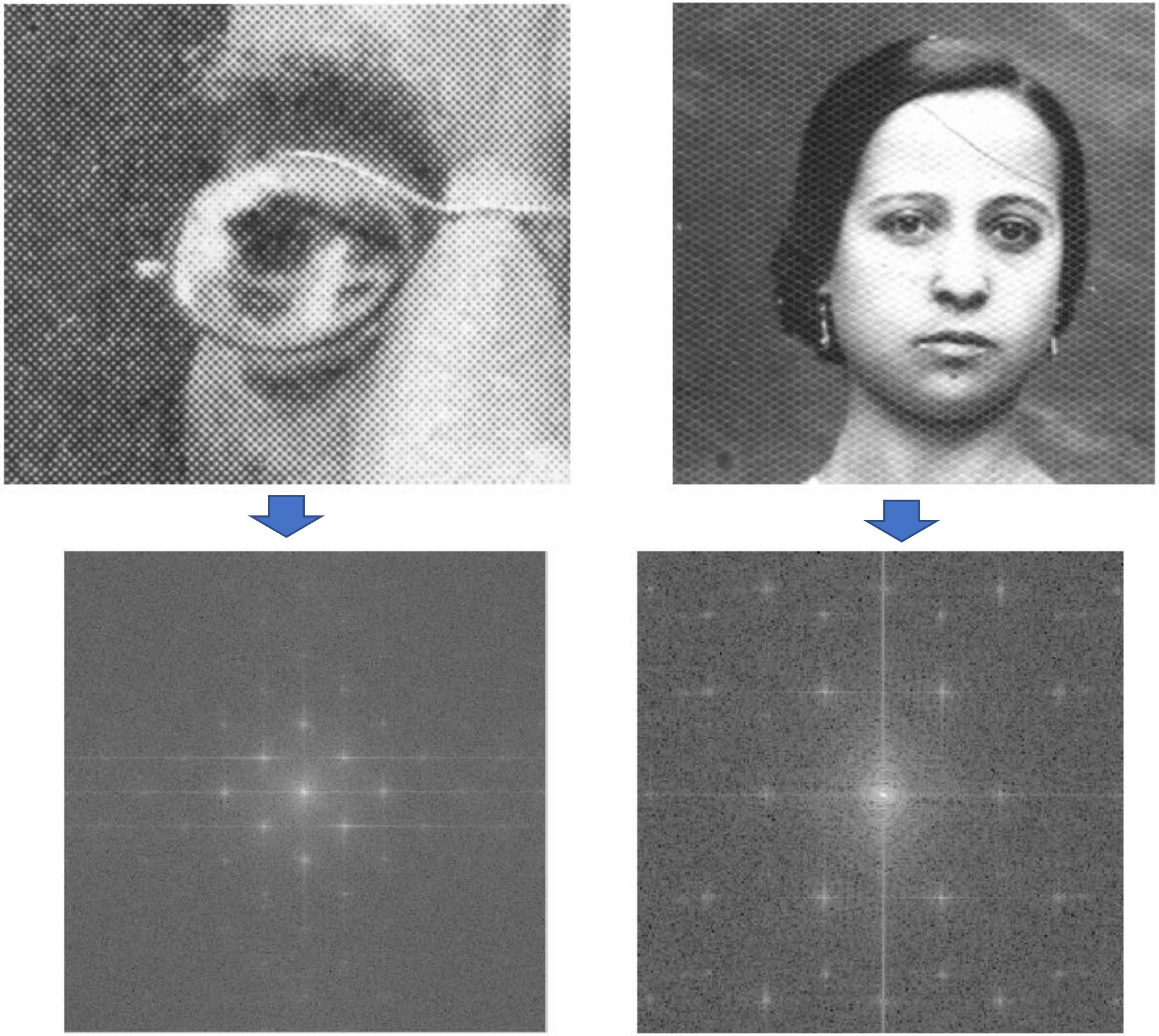
TABLE 4.6

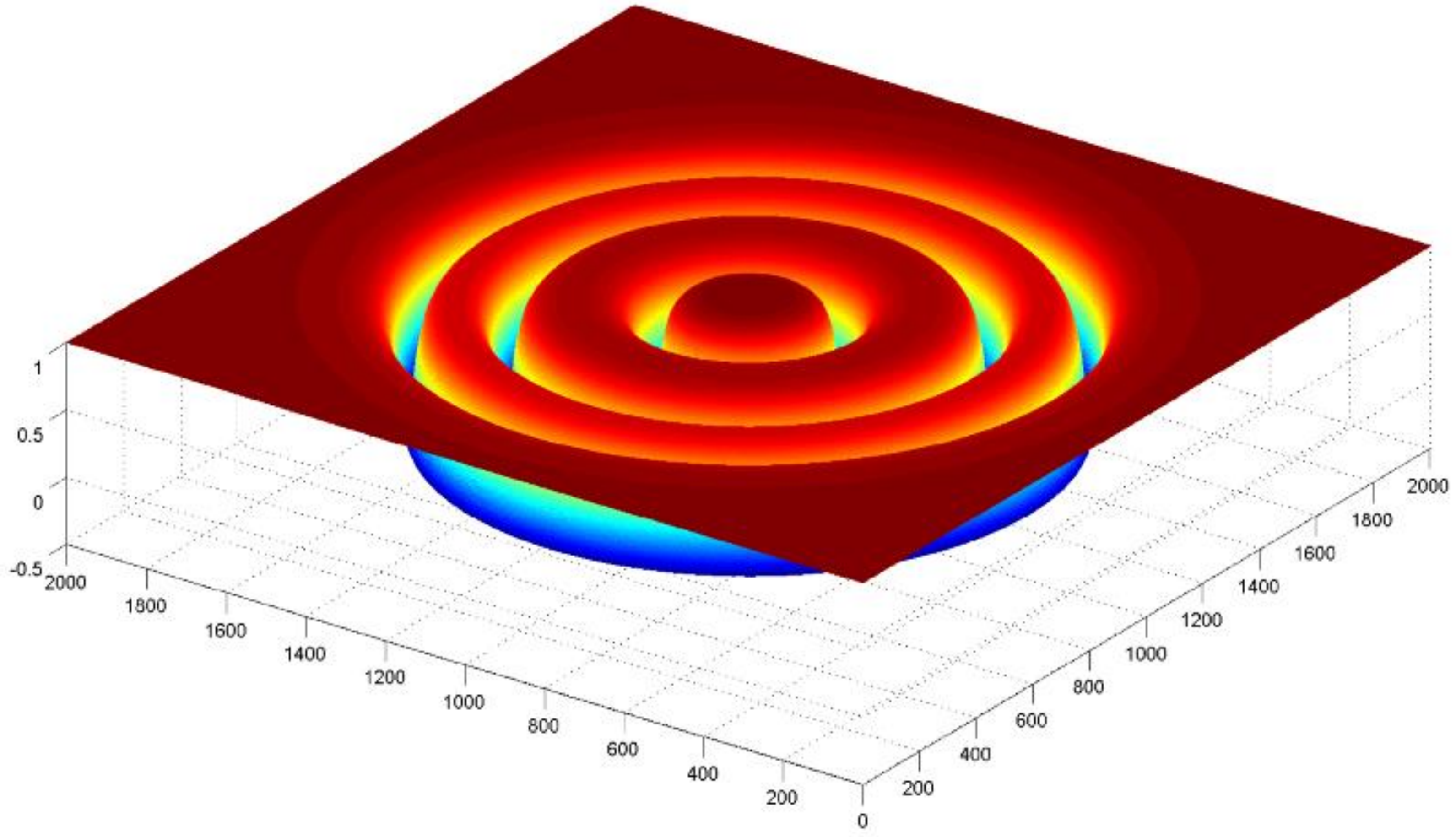
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

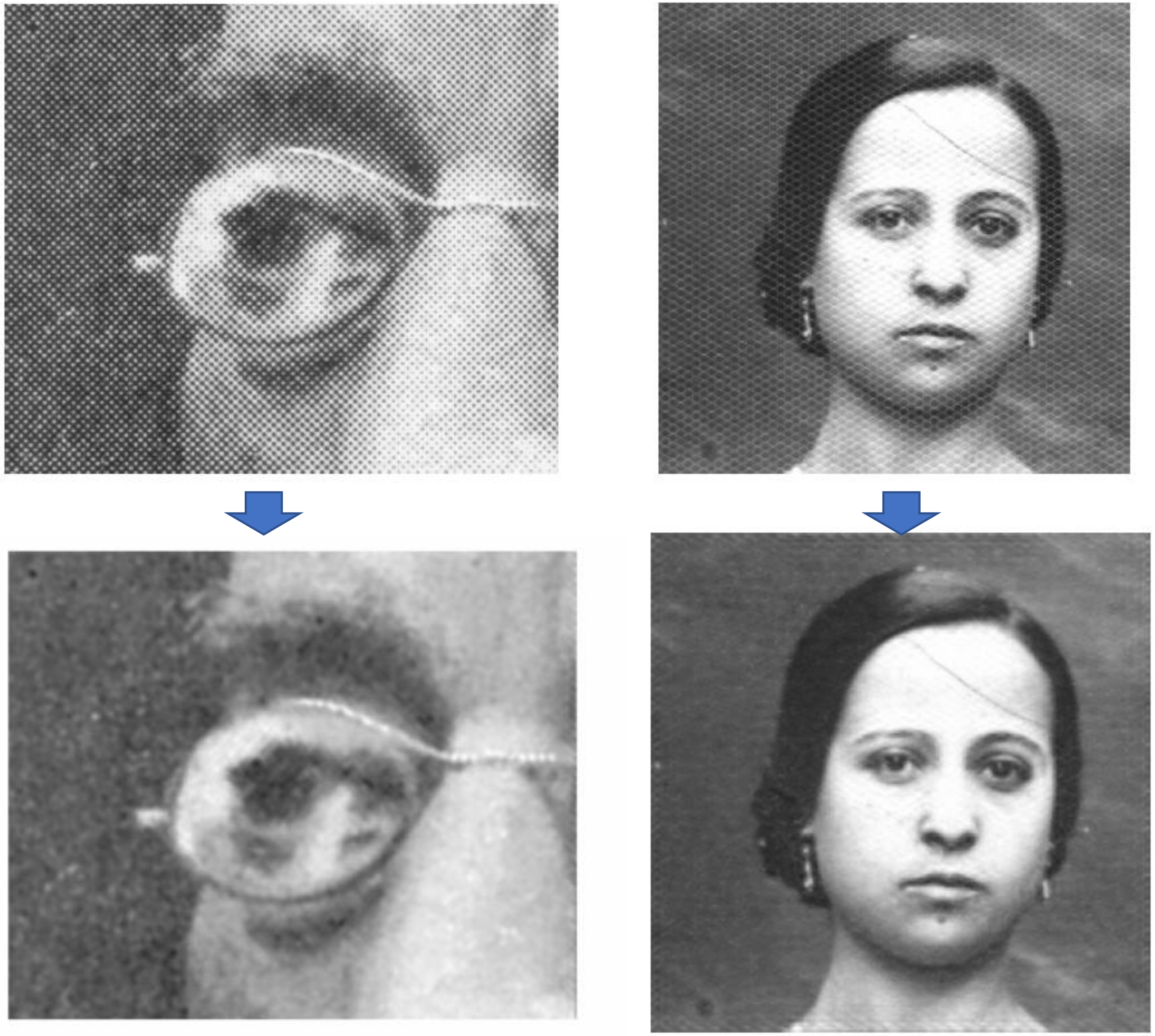
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

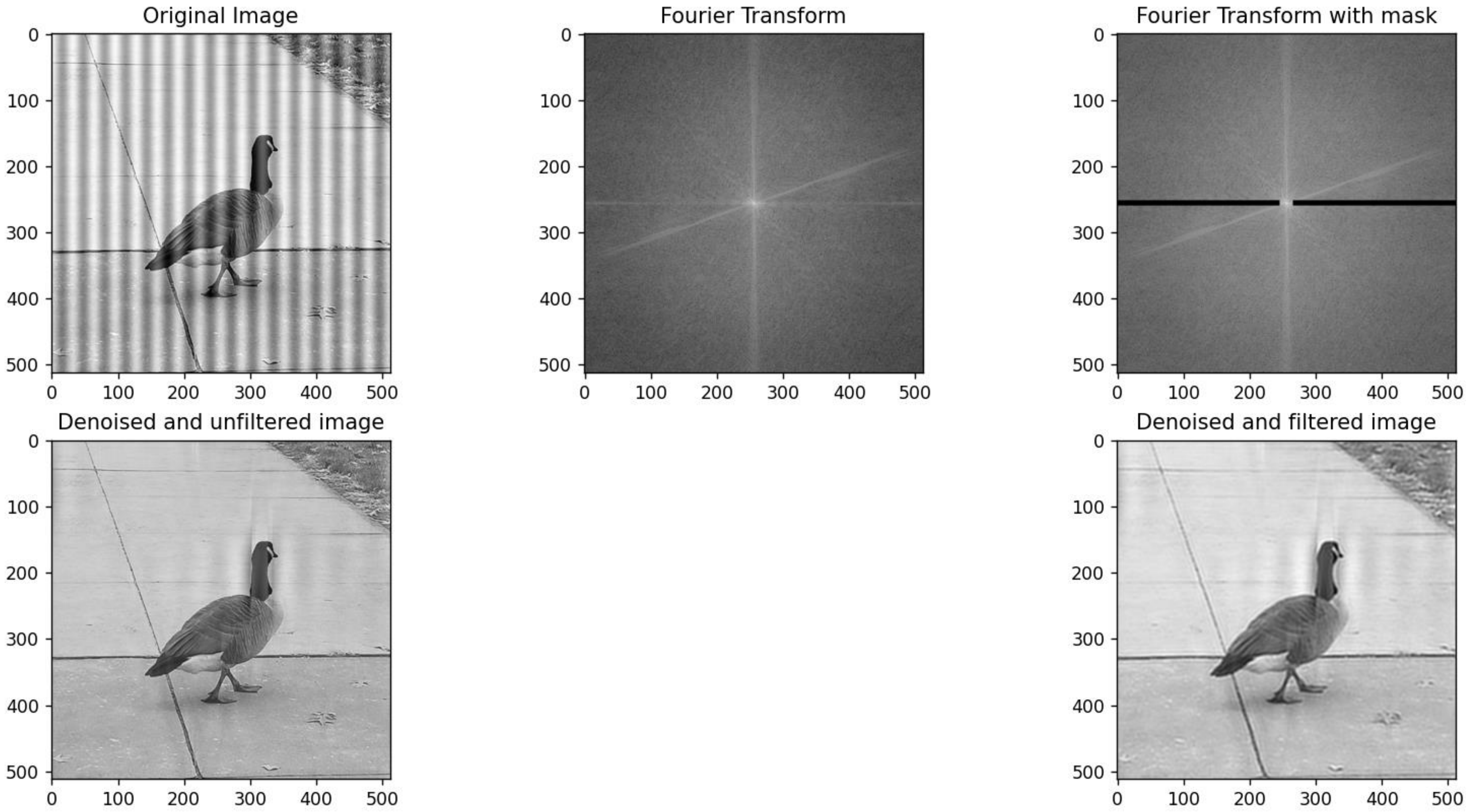
Examples






1
0.5
0
-0.5
2000
1800
1600
1400
1200
1000
800
600
400
200
0
200
400
600
800
1000
1200
1400
1600
1800
2000




Original Image
Fourier Transform
Fourier Transform with mask
Denoised and unfiltered image
Denoised and filtered image
0
100
200
300
400
500

Python code

https://medium.com/@osah.dilshan/applying-fourier-transform-to-images-for-patterned-noise-removal-b543f99f61db

Fourier and the Frequency Domain

Luca Guarnera, Ph.D.

Research Fellow

luca.guarnera@unict.it

University of Catania

Dipartimento di Economia e Impresa